(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 392 556 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
27.06.2012 Bulletin 2012/26

(21) Application number: **02715577.9**

(22) Date of filing: **25.01.2002**

(51) Int Cl.:
B25J 9/16 *(2006.01)* ***B62D 57/032*** *(2006.01)*

(86) International application number:
PCT/GB2002/000338

(87) International publication number:
WO 2002/069256 (06.09.2002 Gazette 2002/36)

(54) ARTIFICIAL MULTIPED AND NEURAL MOTION CONTROLLER THEREFOR

MEHRBEINIGER ROBOTER UND SEINE NEURONALE BEWEGUNGSSTEUERUNG

MULTIPEDE ARTIFICIEL ET SON DISPOSITIF DE COMMANDE DE MOUVEMENT

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR

(30) Priority: **28.02.2001 GB 0104995**

(43) Date of publication of application:
03.03.2004 Bulletin 2004/10

(73) Proprietor: **NaturalMotion Limited**
Oxford OX2 6EE (GB)

(72) Inventor: **REIL, Torsten**
Oxford OX4 1AB (GB)

(74) Representative: **Frost, Alex John et al**
Boult Wade Tennant
Verulam Gardens
70 Gray's Inn Road
London WC1X 8BT (GB)

(56) References cited:
US-A- 5 586 224

- **PATENT ABSTRACTS OF JAPAN vol. 017, no. 675 (M-1526), 13 December 1993 (1993-12-13) -& JP 05 228856 A (FUJITSU LTD), 7 September 1993 (1993-09-07)**
- **JERRY PRATT & GILL PRATT; MIT LEG LABORATORY: "Exploiting Natural Dynamics in the Control of a 3D Bipedal Walking Simulation" INTERNATIONAL CONFERENCE ON CLIMBING AND WALKING ROBOTS, [Online] September 1999 (1999-09), XP002237913 Portsmouth, UK Retrieved from the Internet: <URL:http: //www.coginst.uwf.edu/~jpratt/pu blications/3d_ sim_clawar99.pdf> [retrieved on 2003-04-10]**
- **KUREMATSU Y: "AUTONOMOUS TRAJECTORY GENERATION OF A BIPED LOCOMOTIVE ROBOT" PROCEEDINGS OF THE INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS. SINGAPORE, NOV. 18 - 21, 1991, NEW YORK, IEEE, US, vol. 3, 18 November 1991 (1991-11-18), pages 1983-1988, XP000333213 ISBN: 0-7803-0227-3**
- **JIANG SHAN ET AL: "Design of central pattern generator for humanoid robot walking based on multi-objective GA" PROCEEDINGS. 2000 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS 2000) (CAT. NO.00CH37113), PROCEEDINGS. 2000 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS 2000), TAKAMATSU, JAPAN, 31 OCT.-5 NOV., pages 1930-1935 vol.3, XP002250926 2000, Piscataway, NJ, USA, IEEE, USA ISBN: 0-7803-6348-5**
- **MIYAKOSHI S ET AL: "Three dimensional bipedal stepping motion using neural oscillators- towards humanoid motion in the real world" INTELLIGENT ROBOTS AND SYSTEMS, 1998. PROCEEDINGS., 1998 IEEE/RSJ INTERNATIONAL CONFERENCE ON VICTORIA, BC, CANADA 13-17 OCT. 1998, NEW YORK, NY, USA,IEEE, US, 13 October 1998 (1998-10-13), pages 84-89, XP010311308 ISBN: 0-7803-4465-0**

- **BELLAN Y ET AL: "Artificial neural networks for motion emulation in virtual environments" MODELLING AND MOTION CAPTURE TECHNIQUES FOR VIRTUAL ENVIRONMENTS. INTERNATIONAL WORKSHOP, CAPTECH'98. PROCEEDINGS, MODELLING AND MOTION CAPTURE TECHNIQUES FOR VIRTUAL ENVIRONMENTS. INTERNATIONAL WORKSHOP, CAPTECH '98. PROCEEDINGS, GENEVA, SWITZERLAND, pages 83-99, XP002250927 1998, Berlin, Germany, Springer-Verlag, Germany ISBN: 3-540-65353-8**
- **FUJIMOTO Y ET AL: "SIMULATION OF AN AUTONOMOUS BIPED WALKING ROBOT INCLUDING ENVIRONMENTAL FORCE INTERACTION" IEEE ROBOTICS & AUTOMATION MAGAZINE, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 5, no. 2, 1 June 1998 (1998-06-01), pages 33-42, XP000766217 ISSN: 1070-9932**
- **FUKUDA T ET AL: "Recurrent neural network with self-adaptive GAs for biped locomotion robot" NEURAL NETWORKS,1997., INTERNATIONAL CONFERENCE ON HOUSTON, TX, USA 9-12 JUNE 1997, NEW YORK, NY, USA,IEEE, US, 9 June 1997 (1997-06-09), pages 1710-1715, XP010238720 ISBN: 0-7803-4122-8**
- **CAO M ET AL: "A design method of neural oscillatory networks for generation of humanoid biped walking patterns" ROBOTICS AND AUTOMATION, 1998. PROCEEDINGS. 1998 IEEE INTERNATIONAL CONFERENCE ON LEUVEN, BELGIUM 16-20 MAY 1998, NEW YORK, NY, USA, IEEE, US, 16 May 1998 (1998-05-16), pages 2357-2362, XP010281548 ISBN: 0-7803-4300-X**

Description

FIELD OF THE INVENTION

[0001] This invention relates to computer-generated images of an artificial multiped. The invention relates in particular to a method of controlling the locomotion of a computer generated multiped.

BACKGROUND TO THE INVENTION

[0002] The control of artificial multipeds, whether embodied or physically simulated, is an inherently difficult task. There is considerable interest in this matter, from disciplines as diverse as robotics, computer animation, virtual reality and biology.

[0003] For computer animation, the use of motion capture has long been employed, wherein the desired human behaviour is filmed and then typically digitized to provide data for animation of an equivalent image. Whilst this provides realistic motion dynamics, particularly for the difficult simulation of bipeds, the technique suffers from the difficulty in generalizing walking motions, particularly in unpredictable conditions. The technique also imposes high demands on data storage capacity, and is inappropriate for robots.

[0004] A further approach which has therefore been developed, both for robots and for computer-generated images, relies upon a semi-physical representation of the multiped coupled with a controller to create movement patterns. Techniques such as inverse kinematics and inverse dynamics can be employed to place virtual limbs at the desired positions and accordingly to compute the required forces. The problem with this approach is that the controller tends to use finite state machines (because of the cyclical nature of ambulatory movement) which results in only partially realistic movement and requires hand turning of the parameters of the finite state machine.

[0005] In recent years, neural networks have been employed to improve the control of morpho-functional machines. So called recurrent neural networks (RNNs) have been found to be particularly appropriate for locomotion of multipeds, when used in combination with artificial evolution to optimize the parameter settings of the network. US-A-5,124,918 discusses such an approach in the control of a hexapedal insect-like robot, and Ijspeert et al in "From Lampreys to Salamanders: Evolving Neural Controllers for Swimming and Walking" from Animals to Animats, Proceedings of the 5th International Conference of the Society for Adaptive Behaviour (SAB98), pages 390 to 399, MIT Press (1998) discuss the application of neural networks to artificial salamanders. Golubitsky et al in "Symmetry in locomotor central pattern generators and animal gaits", Nature 401, pages 693-695 (1999) discuss, at a theoretical level only, the neural controllers necessary for many-legged creatures.

[0006] There are particular problems associated with the control of bipeds, both embodied as robots and as computer simulations. Specifically, the unstable dynamics of two-legged walking require that, in general, continuous active control is necessary. Additional controllers have also been necessary to control lateral and sagittal stability. For this reason, engineering techniques such as finite state machines and conventional control theory have been considered necessary. As previously explained, this brings with it the problems of mathematical tractability, the need for manual optimisation, and limits on extendability. Bipeds controlled in this manner also tend to be relatively slow moving.

[0007] "Exploiting Natural Dynamics in the Control of a 3D Biped Walking Simulation", J. E. Pratt et al, International Conference on Climbing and Walking Robots, September 1999, discloses how natural dynamics can be exploited to control bipedal walking robots.

[0008] "Simulation of Autonomous Biped Walking Robot Including Environmental Force Interaction", IEEE Robotics & Automation Magazine, 1 June 1998, discloses an autonomous biped walking control system based on reactive force interaction at the footholds.

[0009] "Artificial Neural Networks for Motion Emulation in Virtual Environments", Y Bellan et al., Proceedings of the International Workshop on Modelling and Motion Capture Techniques, 1998 discloses a data-driven approach to simulation of natural human movement.

[0010] "Autonomous Trajectory Generation of a Biped Locomotive Robot", Proceedings of the International Joint Conference on Neural Networks, 18-21 November 1991, discloses a hierarchical structure for motion planning and learning control of a biped locomotive robot.

SUMMARY OF THE INVENTION

[0011] The present invention seeks to improve on the prior art. In particular, one object of the present invention is to provide an artificial morpho-functional multiped which appears qualitatively more realistic when moving. A further object is to provide a bipedal motion controller which leads to a faster evolution of stable walking, which at least alleviates the problems of the prior art biped controllers and which again causes qualitatively more realistic walking of the biped.

[0012] According to a first aspect of the present invention, there is provided a computer program as defined in claim 1.

[0013] The use of "passive dynamics", that is, the use of freely swinging relatively lower leg parts with only the upper part of the leg being actuated, provides a walking gate of an anthropomorphic nature. It will of course be appreciated that the same effect is achieved even if the lower leg part is actuated, (although at the expense of further hardware and control) provided that the nett result is that the lower leg part moves in a manner which appears to be free swinging or nearly so, and the scope of the claim is to be construed accordingly. Actuation of the lower leg part may in fact be desirable if the multiped is to circumvent or climb over objects. Indeed, a weak constant simulated torque applied between the upper and lower leg parts, which tends to straighten the leg, has been found to be particularly useful. The strength of the torque is sufficiently small that the lower leg part nevertheless appears substantially free-swinging. Likewise, although a leg with only two parts, upper and lower, is preferred, the claim is not restricted thereto and multi-jointed legs are contemplated as well.

[0014] Preferably, the processing means is further arranged to simulate the actuator by calculating the simulated torque, Γ_T, required to move the image of the upper part of each leg, under the effect of the said simulated Newtonian forces, in response to an output from the neural network.

[0015] The actuator used to cause movement of the upper part of each leg may take several forms. In one preferred embodiment, the actuator is a proportional-integral-derivative (P-I-D) actuator in which, preferably, the torque Γ_T at a time T is related to the angle Θ in accordance with the equation

$$\Gamma_T = k_p(\theta_d - \theta) - k_D \frac{d\theta}{dt} + k_I \int_{T-\Delta T}^{T+\Delta T} (\theta_d - \theta)\,dt$$

where k_p, k_D and k_I are the proportional, derivative and integral coefficients, and t is time.

[0016] In particularly preferred embodiments, the coefficient k_I is set to or near zero, so that the actuator becomes a proportional-differential (PD) actuator.

[0017] An actuator implementation of this type requires as input the desired angle between the upper part of the leg and the trunk (equivalent to the desired length of a biological muscle). The necessary force (torque) to reach the "sweet spot" is calculated by the actuator itself. In other words, motor commands from the neural network to the actuator do not, in the preferred embodiment, represent forces, but instead represent desired kinematic parameters like relative joint angles.

[0018] The use of PID or PD controllers brings about several advantages. Firstly, control is provided at a low level, as any discrepancy between the actual and the desired angle is automatically reduced. As a consequence, the tendency of the computer-generated images of the limbs (for example) to buckle under the simulated body mass is counteracted directly by the actuator, a task which would otherwise have to be accomplished by the neural network. In addition, by tuning the allowed desired angle (Θ_d) range, upper and lower limits to limb movement can easily be achieved. This is not possible if the neural network has explicitly to specify forces rather than desired angles.

[0019] Yet a further advantage is that a joint actuator arrangement which acts as one entity rather than implementing single muscles (agonists, synergists, antagonists) greatly reduces the degrees of freedom of the system (and of the neural network controller).

[0020] In other preferred embodiments, however, the processing means is further arranged to simulate the actuator as a force limited velocity constraint actuator in which the torque, Γ_T, applied to the upper part of each of the two legs is a function of the desired velocity of movement of the upper part of the one leg relative to the body trunk, and wherein the desired velocity of movement is related to a difference between the actual and the desired angle between the two upper leg parts, the desired relative angle being derived from the neural network.

[0021] The object point to which an angle of an upper leg part is relative is of course arbitrary for perfectly symmetrical walking. A relative angle between one leg and the trunk is functionally equivalent to a relative angle between one of the upper leg parts a fixed point for example. For perfectly symmetrical walking, indeed, it would be possible to measure the angle of one leg relative to the other.

[0022] In each case, it is to be understood that application of the torque to the upper leg parts may be directly (i.e. to a simulated thigh muscle), to a simulated hip joint or both.

[0023] Preferably, the multiped has only a single actuator per leg, such that the lower leg parts swing freely. Multiple actuators for each leg are however envisaged as well.

[0024] Most preferably, the multiped is simulated as a biped with two hip joints, each hip joint connecting the upper part of a respective one of the two legs to the trunk. The hip joint is constrained to have only a single degree of freedom in the preferred embodiment. It has been found that lateral and twist movements are not necessary for straight line walking.

[0025] Preferably, the processor is further arranged to generate an image of a plurality of feet, each of which appears to be connected to an end of the image of the lower part of a corresponding leg via an ankle joint, and wherein the processor simulates the action of a damped torsional spring to govern the movement of each said ankle joint.

[0026] During walking, the feet represent the only physical interface between a walking multiped and its environment, and therefore play an important role in its locomotion. In particular, by acting analogously to a damped spring, the ankles serve to allow a smooth roll of the foot over the floor whilst simultaneously storing energy. At toe lift-off, this energy is released to used to initiate the next stride. No active neural network control is required for this task.

[0027] The processor may also limit the angle between the image of the upper part and the image of the lower part of each leg. The reason for this is that a passively swinging lower leg preferably requires a mechanism to provide a rapid stop to the movement at full extension. Although an appropriately controlled and timed actuator could in principle achieve this functionality, a preferred solution is to place a soft angular limit on the knee joint (that is, the angle between the upper and lower parts of each leg).

[0028] In a particularly preferred embodiment, the neurones of the neural network are arranged as a recurrent neural network (RNN) so as to form two bi-directionally coupled chains of identical oscillators, at least two of the neurones acting as motor neurones whose output controls the said actuator associated with the upper part of each said leg.

[0029] The neural network architecture of the preferred embodiment acts as a central pattern generator. The number of identical oscillators in each chain is preferably twice that of the appendages to be actuated. In other words, four oscillators per chain or eight in total are preferred to control quadrupeds, whereas two oscillators per chain, i.e. four in total, are desirable for bipeds.

[0030] Most preferably, the recurrent neural network comprises one recurrent subnet per oscillator, and each recurrent subnet preferably comprises six nodes as this was found to produce a high degree of spontaneous cyclic activity. Each of these recurrent subnets is connected to a laterally adjacent recurrent subnet (i.e., a recurrent subnet in the other of the two chains) via symmetrical weights so as to allow bidirectional coupling. Each neurone in each recurrent subnet is preferably arranged to synapse with its equivalent in the opposite recurrent subnet in the opposite chain. Connections to sagittal neighbours are asymmetric, in the preferred embodiment, that is, each neurone is preferably connected to its equivalent in the next recurrent subnet in the same chain, but no connections exist the other way.

[0031] For a biped, preferably, two of the nodes in the RNN are designated as motor neurones, with the output of these nodes being used to control left and right sagittal hip actuation. More generally, it is preferable to have one motor neurone per leg to be actuated in a multiped.

[0032] The use of identical oscillators as well as only two sets of coupling weights (lateral and sagittal) limits the search spaced (for the network parameters) to a total of 60 parameters (36 weights plus 12 neurone parameters plus 6 lateral weights plus 6 sagittal weights).

[0033] Furthermore, an ancillary stability controller may be applied at the beginning of an optimization procedure, to improve the overall speed of evolution of stable network parameters. The effect of this ancillary controller is gradually reduced in subsequent optimisation iterations.

[0034] Preferably, the properties of the neurones and their connections within the said neural network are determined on the basis of a genetic algorithm. Most preferably, the criterion used to optimise the network parameters is that the distance over which an end of the lower part of a rearmost leg in the image of the multipedal object, relative to that origin, should be maximized relative to the said origin.

[0035] Rapid optimisation of the network parameters is further achieved, in preferred embodiments, by aborting consideration of trial arrays of networks parameters as soon as certain undesirable situations occur. For example, if the centre of mass of the simulated body drops below a certain, predetermined value, then any resultant locomotion will certainly not appear realistic and the network parameters resulting in this form of image control are considered non-ideal. Enforcing the early abortion of unpromising runs speeds up evolution dramatically.

[0036] In preference, the computer program may be stored upon a computer-readable storage medium such as a magnetic or optical disc.

[0037] According to a second aspect of the present invention, there is provided a method of controlling the locomotion of a computer-generated multiped as defined in claim 24.

[0038] Although aspects of the invention focus upon neural control of bipeds, it is to be understood that the invention has broader application.

[0039] Preferred aspects of the invention thus provide several advantages: a simulated biped is provided in which the length of evolutionary optimization required to evolve parameters for a neural network that produces straight line walking of that biped is very short. Moreover, the large majority of neural networks obtained via the optimization procedures result in successful controllers. Furthermore, the perceived realism of the body dynamics of a biped or other device/ object having more than two legs is increased relative to actual multipedal walking, particularly for bipeds such as humans. This can be confirmed by comparison with motion capture data. The use of free swinging lower legs constrains the walking dynamics in a particularly favourable manner. Walking styles that do not make use of a full lower leg swing (and hence a kneecap) were less stable than those which do.

[0040] The neuro-evolutionary approach of preferred aspects of this invention allows the overall body control to be fully automated which provides several advantages, including body implementation flexibility and the avoidance of the need for manual retuning of controller parameters. The neural network architecture of further aspects of the invention also allows the integration of sensory (in particular proprioceptive) feed back and tonic input to the controller, which may in turn allow improved robustness of walking gaits on uneven terrain by appropriate modulation of the neural network.

BRIEF DESCRIPTION OF THE DRAWINGS

[0041] The invention may be put into practice in a number of ways, and one embodiment will now be described by way of example only and with reference to the accompanying drawings in which:

Figures 1a and 1b show fully rendered and wire-frame versions of a computer-generated biped embodying an aspect of the present invention;
Figures 2a-2d show side views of a sequence of movement of the biped of Figure 1a;
Figure 3 shows a schematic arrangement of a central pattern generator (CPG) suitable for controlling the locomotion of the biped of Figures 1 and 2, and embodying another aspect of the present invention;
Figure 4 shows a specific neurone arrangement to implement the CPG of Figure 3 as a neural network;
Figure 5 shows a decision tree for evolving the parameters of the neural network of Figure 4, for example;
Figures 6a and 6b show two exemplary graphs of fitness, defined as distance travelled by the rear foot of the biped of Figures 1 and 2, as a function of the number of generations of evolution of the neural network parameters, when evolved using the decision tree of Figure 5 and when applied to the neural network of Figure 4;
Figures 7a and 7b show two further exemplary graphs of fitness versus number of evolutionary generations, with an ancillary stabilisation controller employed that is deactivated after the fitness reaches a predetermined distance;
Figure 8 shows another exemplary graph of fitness versus the number of evolutionary generations, with no ancillary stability controller employed;
Figure 9 shows the output, in radians, of the neural network of Figure 4 with optimised parameters, and the resulting body dynamics of the biped of Figures 1 and 2, also in radians, each as a function of time;
Figure 10 shows a plot of the left hip angle of the biped of Figure 1 as a function of the left hip motor neurone output in the neural network of Figure 4;
Figures 11a and 11b show attractor plots of left and right hip neurone outputs, and left and right actual angles, respectively, for the biped of Figures 1 and 2;
Figure 12 shows a plot of the left hip angle of the biped of Figures 1 and 2, when actuated in accordance with one aspect of the present invention, as a function of the left knee angle of the biped of Figures 1 and 2; and
Figure 13 shows, in perspective, a further sequence of movement of the biped of Figures 1 and 2 when actuated in accordance with an aspect of the present invention and controlled in accordance with another aspect of the present invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

[0042] The following will describe, by way of example, the currently preferred arrangement which illustrates the various aspects of the present invention. The bipedal object in the following example is computer-generated. However, it is to be understood that the principles underlying the present invention are equally applicable to quadrupeds and other multi-limbed creatures.
[0043] Referring first to Figures 1a and 1b, fully rendered and wire-framed versions of a computer-generated biped 10 are respectively shown. The biped is simulated using a rigid body dynamics simulation program. Such programs are well known in the art and are available either as commercial or as freeware/open source software. Rigid body dynamics simulation software simulates the motion of idealised bodies connected by idealised joints. The bodies are able to collide with one another, rest and slide over one another and so forth. The software calculates the Newtonian mechanics for a body having specified parameters, so that when these bodies are moved, they appear to move as if under the influence of gravity, for example.
[0044] The biped shown in Figure 1a in particular is simulated using Dynamics Toolkit Version 0.0.5, by MathEngine plc, Oxford, UK. This program is particularly preferred because it is optimised for speed rather than accuracy which is useful when using evolutionary optimisation. A further advantage is that this program includes a force limited velocity constraint algorithm. This is useful for actuation of the limbs as will be explained below.
[0045] The MathEngine program is delivered as a suite of libraries coded in C and assembler. Objects are called from the libraries to build the simulated biped shown in Figures 1a and 1b.
[0046] As seen more clearly in Figure 1b, the biped which is simulated by the MathEngine program comprises a body trunk 20 and two legs 25 which each depend from the body trunk 20. Each leg comprises a first, upper part 35 and a second, lower part 45. The upper part 35 of each leg is journalled about the body trunk 20 via a corresponding hip joint 30. The lower part 45 of each leg pivots about the upper part 35 of the same leg via a knee joint 40. At the end of each leg is a foot 60 which pivots about the lower part 45 of an associated leg via an ankle joint 50. In currently preferred embodiments, the biped has eight degrees of freedom: hip-roll, hip-pitch, knee-pitch, and ankle-pitch. As will be explained

below, only the two hip joints 30 are actively actuated.

[0047] The specific dimensions of the biped shown in Figures 1a and 1b are set out in Table 1 below.

TABLE 1

BODY	IMPLEMENTATION	DIMENSIONS
Heel	sphere	0.08m radius
Toe	sphere	0.08m radius
Calf	cuboid	0.125m x 0.540m x 0.125m
Thigh	cuboid	0.175m x 0.500m x 0.175m
Hip	sphere	0.125m radius
Pelvis/body trunk	cuboid	0.500m x 0.500m x 0.600m

[0048] Each body part, as set out in Table 1 above, is modelled using a simulated density of 1kg/l. It will be appreciated that a simple block is employed to simulate the body trunk of the biped in the current implementation. This is because the dynamics of the upper part of the body present different (and, generally, more straightforward) problems to those which are addressed by the present invention. Therefore, to aid understanding, the details of the upper part of the body of the biped described herein are not given.

[0049] In order to allow simulated walking of the biped shown in Figures 1a and 1b, three distinct design solutions are proposed. Each individually and in combination facilitates the rapid evolution of an anthropomorphic walking style.

Body Dynamics

[0050] The biped of Figures 1a and 1b is caused to move by a combination of actuation and passive swinging. In particular, the hip joints 30 actuate the upper leg parts 35, and the lower leg parts 45 are allowed to follow passively, i.e., without separate actuation. Given this, the control task for straight line walking of the biped 10 can be simplified to two degrees of freedom (the hip joints 30) for the biped 10.

[0051] The dynamics of a leg 25 in which the upper part 35 is actuated and the lower part 45 swings passively are shown in the sequence of Figures 2a to 2d. The passive, lower leg part 45 is accelerated only by the effect of the simulated gravity thereupon, in combination with the pulling movement of the upper leg part 35. In addition to simplification of the control (because the knee joint is not actively actuated, in contrast to previous artificial morpho-functional bipeds), this implementation markedly increases the realism of the walking pattern produced.

[0052] Because the lower leg part 45 of each leg 25 is allowed to swing passively, a mechanism is required to provide a rapid stop to the movement at full extension. Whilst an appropriately controlled and timed actuator could in principle achieve this functionality, it is simpler to employ a simulated kneecap. In the preferred implementation, a soft angular limit to the knee joint 40 is simulated.

[0053] Whilst a completely free-swinging kneecap is successful in allowing realistic walking gaits to be simulated, in fact in presently preferred embodiments a weak constant torque is applied between the upper and lower leg parts 35, 45. The torque is applied in the direction that would tend to straighten the leg. It is to be understood that this weak constant torque is by no means essential, however, and in any event is sufficiently weak that it is not, or at least barely, discernable to the human eye.

[0054] The simulated biped 10 also employs feet 60. These represent the only physical interface between a walking biped and its environment, and therefore play an important role in its locomotion. By acting analogously to a damped spring, the ankle joints 50 serve to allow a smooth roll-off of the foot 60 over the simulated floor, whilst simultaneously storing energy. Upon liftoff of the front of the foot, this energy is released and used to initiate the next stride. No active control of the foot 60 or ankle joint 50 is required for this task. The presently preferred implementation simulates a damped torsional spring in each of the ankle joints 50. Again, this can readily be performed through the libraries available in the MathEngine program.

Actuators

[0055] In humans and animals, simple lower level mechanisms such as spinal reflexes are believed to be important in coordination of movement. In the present case, an actuator is employed as a low level controller which takes as inputs the relative angles and angular velocities of the two body parts (here, the upper leg parts 35) being actuated from a simulation, a desired relative angle from the neural controller, and which calculates a torque to be applied equally and oppositely to the two body parts being actuated.

[0056] An actuator must generate the effect of turning body parts towards the desired relative angle smoothly and at a reasonably controlled pace.

[0057] There are several types of actuator that can be used, and which provide the level of control necessary for stable biped locomotion. For example, a proportional integral derivative (PID) controller can be used. Applied to the actuator control of the hip joints in the biped 10 of Figures 1 and 2, it is possible to express the torque T as a function of the desired angle between the upper part 35 of the leg and the body trunk 20 and the actual angle between those two parts. In particular, the torque Γ_T at time T can be expressed as

$$\Gamma_T = k_P(\theta_d - \theta) - k_D\frac{d\theta}{dt} + k_I\int_{T-\Delta T}^{T+\Delta T}(\theta_d - \theta)dt$$

where k_p, k_D and k_I are the proportional, derivative and integral coefficients respectively, Θ_d is the desired angle and Θ is the actual angle. In practice, the integral coefficient k_I and the period t are small or vanishing. Thus the actuator tends to a proportional derivative actuator.

[0058] The code to simulate the actuator is written in C or C++ and forms a subroutine called from within MathEngine.

[0059] An actuator implementation of this type requires as input the desired angle which is, in this case, equivalent to the desired length of a biological muscle. The necessary force to reach the "sweet spot" is calculated by the actuator itself. That is, motor commands from higher control centres (such as a central pattern generator, as described below) to actuators do not represent forces, but desired kinematic parameters like relative joint angles.

[0060] The use of PID controllers brings about several advantages. They provide control at a low level, as they automatically reduce any discrepancy between the actual and the desired angle. As a consequence, the tendency of limbs, for example, to buckle under the simulated body mass is counteracted directly by the actuator, a task which would otherwise have to be accomplished by a separate controller. In addition, by tuning the allowed desired angle (Θ_d) range, upper and lower caps to limb movement can easily be achieved. This is not possible if the higher control centre is required explicitly to specify forces rather than desired angles.

[0061] Finally, a joint-actuator arrangement that acts as one entity rather than implementing single muscles (agonists, synergists, antagonists) greatly reduces the degrees of freedom of the system and of the overall system controller (for which, again see below).

[0062] Whilst PID or PD controllers have their advantages, they do also suffer from a number of drawbacks. Firstly, changes in the control environment (e.g. the simulated load on the simulated body) require dynamic parameter modulation of the PID. Because this modulation must itself be controlled, more degrees of freedom are inevitably added to the system.

[0063] Secondly, PID controllers may bring about numerical instabilities. For these reasons, the presently preferred implementation employs a different but conceptually analogous actuator type known as a force limited velocity constraint actuator. Such actuators allow the user to specify a desired relative velocity between two bodies. They tend to be more reliable than PD or PID actuators, but are typically computationally more expensive.

[0064] In the present case, the relative angle between the two upper leg parts 35 is used as a first input. A desired angle is input from the neural controller (see below). A demand velocity is then calculated whose magnitude is related to the difference between the actual relative angle and the desired relative angle. The demand velocity is used by the constraint solver of the simulation to calculate the torques required to satisfy this request. The torque is applied equally and oppositely to the two upper leg parts 35, unless the calculated torque exceeds a maximum (Γ_{max}), in which case the maximum is applied. In that case, then, an external modulation of controller parameters becomes unnecessary.

[0065] In the case of each actuator type, several constants need to be predetermined, i.e. k_P, k_D, k_I, Δt or Γ_{max}. These are determined by experimentation with the simulation and can be used from then on for all simulations of similar scale.

Controller Architecture

[0066] The third part of the system which allows realistic locomotion of the biped 10 is a neural network which acts as a global controller for the biped. The structure is shown in Figure 3. The structure comprises four identical oscillators 70, 80, 90, 100, arranged as two bi-directionally coupled chains. More generally, the number of oscillators in each chain should correspond to the number of appendages to be actuated. Thus, for quadrupeds, each chain would consist of four identical oscillators such that the overall structure contains eight identical oscillators.

[0067] The generalised structure of Figure 3 is implemented for the computer-simulated biped 10 as a neural network, again written as a subroutine in C or C++; the network connections are shown in Figure 4. Each oscillator 70, 80, 90, 100 is represented by a fully connected recurrent neural network having six nodes 71-76, 81-86, 91-96 and 101-106. The oscillators 70 and 90 are laterally adjacent but in different oscillator chains. Likewise, the oscillators 80 and 100 are laterally adjacent but in different oscillator chains. Each neurone in one of the laterally adjacent oscillators is connected to its equivalent neurone in the laterally opposite oscillator via a symmetrical weight so as to allow bi-directional coupling. For example, the neurone 71 in the oscillator 70 is connected to the neurone 91 in the oscillator 90.

[0068] By contrast, the connections to the sagittal neighbours are asymmetric, as seen most readily in Figure 3. In the neural network arrangement of Figure 4, each neurone in oscillator 80 (such as neurone 86) is connected to its equivalent (neurone 76 in this example) in the next or other oscillator in the same chain (in this case, oscillator 70), but no connections exist the other way.

[0069] The neurone model of Figure 4 includes a time constant τ_j and a bias t_j.

[0070] At each iteration, the activity of the j^{th} neurone in the RNN of Figure 4 is computed according to:

$$\tau_j \dot{A}_j = -A_j + \sum \omega_{ij} O_i$$

where τ_j is the time constant of the j^{th} neurone, A_j its activity, O_i the output of the i^{th} to the j^{th} neurone, and ϖ_{ij} the weight from the i^{th} to the j^{th} neurone. The corresponding output is:

$$O_j = (1 + e^{(t_j - A_j)})^{-1}$$

where t_j is the bias of the j^{th} neurone.

[0071] These parameters have allowed ranges, as do the synapse weight parameters.

[0072] The ringed neurones 81 and 101 in Figure 4 are designated motor neurones. Their outputs are between 0.0 and 1.0 and are scaled to drive the actuators to move the upper leg parts 35 through the allowed angles. The motor neurone 101 is used for sagittal actuation of the left hip joint 30 of the biped 10, and the motor neurone 81 is used for sagittal actuation of the right hip joint 30 in the biped 10.

[0073] The use of identical oscillators as well as only two sets of coupling weights (lateral and sagittal) limits the search space to a total of 60 parameters (36 weights, 12 neurone parameters, 6 lateral weights and 6 sagittal weights). This allows the time taken to optimise the parameters to be reduced dramatically.

<u>Genetic Algorithm</u>

[0074] The parameters of the neural network which are to be evolved are the synapse weights and the node properties (time constants and biases). These are encoded as real numbers in a linear chromosome with different ranges for each parameter, typically [-16.0, 16.0] for the weights, [0.5, 5.0] for the time constants and [-4.0, 4.0] for the biases. The algorithm as a whole is yet another bespoke C or C++ subroutine which can be called as necessary by the graphics simulator. The mutation rate is chosen so as to cause an average of one substitution per individual. No crossover operator is applied.

[0075] Rank-based selection is used on a population of 100 individuals, with a fittest fraction of 0.5. This is equivalent to culling the bottom, weakest half of the population, and replacing it with a copy of the top half. This latter procedure is known, as such, from, for example, W.D. Hillis, "Co-evolving parasites improves simulated evolution as an optimisation procedure", in Artificial Life II, edited by C. Langton et al, Addison-Wesley at pages 313-324.

[0076] The individuals are selected on the basis of a fitness function which is depicted graphically in Figure 5. The overall fitness criterion selected is the distance, d, of the hindmost foot 60 of the simulated biped 10 from an origin at which locomotion commences. The hindmost foot is chosen as a reference point so as to penalise individuals that take only one large stride.

[0077] The fitness function decision tree of Figure 5 serves several functions. Importantly, it enforces the early abortion of unpromising runs ($t < t_e$), in order to speed up evolution. A run is aborted if the position of the centre of mass (CoM) falls below a predetermined height, β, i.e. $h<\beta$. This criterion penalises falling down, as well as grotesque movements. Individuals that fall over without previously moving ($d<\alpha$, where α is a distance threshold) are assigned a fitness of 0. If a point attractor ($\Delta a=0$) is reached, and the biped's movement forward stops, then the result may be discarded (i.e. fitness set to zero) or may be used as part of the next population (d_t), at the user's discretion. In Figure 5, t is time, t_e is maximum time, h is the CoM height, β is the CoM height threshold, α is a distance threshold, Δa is a measure of the change in activation in the neural network of Figure 4, and d is the distance from the hindmost foot 60 of the biped 10 to an origin.

[0078] It has been found that a weak assisting controller is preferable to permit rapid evolution of stable locomotion control of the biped 10. This assisting controller is applied only at the beginning of an evolutionary run, in order to support the trunk 20 both laterally and sagittally. The controller takes the form of an un-damped orientation spring between the trunk 20 and a reference "frame" for the biped 10. The spring has no displacement parameters, that is, the torque exerted on the trunk 20 is dependent only on its orientation, not its position.

[0079] The ancillary controller serves initially to ' guide evolution towards the production of rhythmic step cycles during early mutations. When the genetic algorithm has found a solution with suitably high fitness using the torso stabiliser, its strength (that is, the spring stiffness) is reduced and the search restarted, seeded with the population from the previous optimization procedure. The stiffness of the spring is iteratively reduced until the stabiliser is weak enough to be discarded or at least not to visibly impair biped simulation. The advantage of this procedure will be further understood with reference to Figures 7a, 7b and 8 described below.

[0080] Referring now to Figures 6a and 6b, two representative plots of fitness, in metres, as a function of number of generations of evolution are shown. It is noted that each of these representative plots indicates that successful straight line walking has been achieved. In each case, the ancillary controller described in connection with Figure 5 was activated. It will in particular be noticed that realistic straight line walking occurs within a relatively small number of generations. The average walking speed of walkers was 0.8 m/s with maximum speeds of about 1.2 m/s reached by some individuals.

[0081] Because initial runs were performed with a weak stabilising ancillary controller, populations of controllers that permitted stable walking of the biped 10 were picked at random from a series of evolutionary runs and re-evolved with the effect of the ancillary controller substantially reduced. It was found that these individuals reliably evolved to attain the previously achieved fitness. This is shown in Figures 7a and 7b. Here, runs were evolved with ancillary controllers until the biped reached a distance of 20 metres from the origin. The ancillary controller was then switched off and the population re-evolved.

[0082] By contrast, control experiments with de novo evolution and de-activated or at least substantially reduced ancillary controllers did not result in successful runs, the measure of success being defined in this case as a minimum fitness of 20 metres. This may be seen from Figure 8.

[0083] The analysis of the neural activity and the resulting body dynamics show that relatively simple periodic controller outputs sufficed to produce stable walking. This may be seen from Figure 9, which shows a plot of neural output (desired angles for left and right hips 50, in radians) as a function of time, as well as the resulting body dynamics (the actual angles of the left and right hips 50, again in radians), as a function of time. It will be noted that (a) the periods of the neural output, and the body dynamics (such as the hip angles) were locked, and (b) there is a small sub-oscillation in the neural output. This sub-oscillation is, however, not reflected in the body dynamics, which are shown in Figure 13 (from which the data of Figure 9 were obtained). This is because the lag between actual and desired angles acts as a low pass filter, smoothing out small oscillations in the desired angle trace.

[0084] Figure 10 shows a plot of the angle of the left hip 50 of the biped 10 as a function of the output of the motor neurone 101 (Figure 4) which controls left hip actuation. It will likewise be seen from this Figure that, after an initial settling phase, both periods become locked to one another.

[0085] Figure 11a shows an attractor plot of the desired angles of the left and right hip joints 50. Figure 11b shows an attractor plot of the actual angle of the left and right hip joints 50. Periodic gaits tend to be preceded by a short settling phase, in which the neural outputs and the body dynamics differ from the subsequently established pattern. By visual analysis, it would appear that this initial settling phase is used to initialise the body position for the walking gait. The initial settling phase is more prominent where passive body dynamics are employed (as described in connection with Figure 2 in particular), rather than a fully actuated leg. This is best understood from Figure 12, which shows the angle of the left upper leg part 35 of the biped 10, relative to the body trunk 20, as actuated by the hip joint 30, as a function of the angle of the lower leg part 45 relative to the upper leg part 35, which lower leg part is not actuated. It will be noted that it takes several steps until the knee joint 40 reaches its final full swing cycle.

[0086] Figure 13 shows screen shots of the biped 10 of Figures 1 and 2, in perspective view, over one gait cycle. It will be seen even from the ten frames illustrated that the gait is highly anthropomorphic i.e. realistic, making full use of the passive dynamics of the body such as the free-swinging lower legs.

[0087] Although one specific embodiment has been described, the skilled person will readily understand that various modifications or improvements may be made without departing from the scope of the invention as defined by the accompanying claims. For example, although the foregoing description and drawings refer to a specific rigid body dynamic simulator, it will be understood that the specific implementation is not crucial to an understanding of the invention. For example, other simulation tools, such as FastDynamics Version 1.005, or the more recent "Dynamics Toolkit", both by MathEngine plc could equally be used. Havok Hardcore™, by Havok and Vortex™, by Critical Mass Labs, are likewise suitable physics engines. Furthermore, because of the increased flexibility afforded by the neuro-evolutionary approach outlined above, in particular with respect to body implementation, it is relatively straightforward (and quick, on a modern p.c.) to test different mass distributions, body dimensions, and actuator or joint types simply by evolving suitable controllers.

Claims

1. A computer program comprising computer program elements which, when executed on a digital computer, cause processing means thereto:

   (i) to generate an image, on a visual display unit, of a multiped (10) having a trunk (20) and at least two legs (25) depending therefrom, each leg including at least a first, upper part (35) and a second, relatively lower part (45);
   (ii) to simulate the Newtonian forces that would be acting upon each separate part of the multiped at a given time and to apply these simulated forces to each said separate part so that, when displayed, each part moves under the influence of these simulated forces;
   (iii) to cause the image of each upper leg part to pivot about the trunk;
   (iv) to cause the image of the upper part of each leg to be moved via a simulated actuator in an arc relative to the said trunk;
   (v) to cause the image of each relatively lower leg part to describe an apparently substantially free-swinging movement relative to its respective upper part; and
   (vi) to provide a neural network having a plurality of neurones, the output of which neural network controls the actuator movement of each upper leg part such that the image of the multiped appears to walk.

2. The program of claim 1, further arranged to cause the processing means to simulate the actuator by calculating the simulated torque, Γ_T, at a time T that is required to move the image of the upper part of each leg, under the effect of the said simulated Newtonian forces, in response to an output from the neural network.

3. The program of claim 2, in which the actuator is a proportional derivative actuator, wherein the simulated torque, Γ_T, applied to the upper part of each leg is a function of the angle, Θ, of the upper part of each said leg relative to another body part.

4. The program of claim 2, in which the actuator is a proportional-integral-derivative actuator wherein the simulated torque, Γ_T, applied to the upper part of each leg is a function of the angle, Θ, of the upper part of each said leg relative another body part.

5. The program of claim 4, in which the torque Γ_T, at time T is related to the angle Θ in accordance with the equation

$$\Gamma_T = k_P(\theta_d - \theta) - k_D\frac{d\theta}{dt} + k_I\int_{T-\Delta T}^{T+\Delta T}(\theta_d - \theta)dt$$

   where k_P, k_D and k_I are the proportional, derivative and integral coefficients, and t is time.

6. The program of claim 1, further arranged to cause the processing means to simulate the actuator as a force limited velocity constraint actuator in which the torque, Γ_T, applied to the upper part of each of the two legs is a function of the desired velocity of movement of the upper part of a leg relative to the body trunk,
   and wherein the desired velocity of movement is related to a difference between the actual and the desired angle between each upper leg part and the body trunk, the desired relative angle being derived from the neural network.

7. The program of any one of the preceding claims, further arranged to cause the processing means to generate a simulated hip joint for each leg, to link the upper part of a given leg to the trunk and to simulate at least one actuator per leg, each actuator being arranged to act upon the hip joint associated therewith so as to cause the image of the corresponding upper leg part to move.

8. The program of claim 7, further arranged to cause the processing means to simulate one actuator per leg.

9. The program of claim 7 or claim 8, further arranged to cause the processor to constrain the simulated hip joint to have only one degree of freedom such that the image of the corresponding upper leg part is constrained to move only sagitally.

10. The program of any one of the preceding claims, further arranged to cause the processing means to generate an image of a plurality of feet, each of which appears to be connected to an end of the image of the lower part of a

corresponding leg via an ankle joint, and wherein the processing means simulates the action of a damped torsional spring to govern the movement of each said ankle joint.

11. The program of any one of the preceding claims, further arranged to cause the processing means to limit the angle between the image of the upper part and the image of the lower part of each leg.

12. The program of any of the preceding claims, in which the neurones of the neural network are arranged as a recurrent neural network (RNN) so as to form two bi-directionally coupled chains of identical oscillators, at least two of the neurones acting as motor neurones whose output controls the said actuator associated with the upper part of each said leg.

13. The program of claim 12, in which the RNN comprises a plurality of neurones arranged so as to form two bi-directionally coupled pairs of identical oscillators.

14. The program of claim 13, in which the RNN comprises one recurrent subnet per oscillator, each of which has six nodes.

15. The program of claim 14, in which each recurrent subnet in one of the chains is connected to a laterally adjacent recurrent subnet in the other of the chains via symmetrical weights so as to provide bi-directional coupling therebetween, and in which each neurone in each recurrent subnet is arranged to synapse with an equivalent neurone in the said laterally adjacent recurrent subnet.

16. The program of claim 15, in which each neurone within each recurrent subnet in one of the chains is asymmetrically (unidirectionally) connected to a corresponding neurone within a next recurrent subnet in the same chain.

17. The program of any one of the preceding claims, further configured to cause the processing means to determine the properties of the neurones and their connections within the said neural network on the basis of a genetic algorithm.

18. The program of claim 17, further configured to cause the processing means to determine the said properties on the basis of a genetic algorithm in which the said properties are optimized by the identification of a fittest array of property values from a plurality of different arrays of property values in accordance with the criterion that the distance over which an end of the lower part of the rearmost leg in the image of the multiple, relative to an origin, should have travelled a maximum distance from the said origin.

19. The program of claim 18, further configured to cause the processing means to abort the consideration of a given one of the arrays of property values if it is determined that a point attractor has been reached such that the image of the multiped stops moving away from the origin.

20. The program of claim 18 or claim 19, further configured to cause the processing means to abort the consideration of a particular one of the arrays if it is determined that the centre of mass of the trunk of the multiped, as calculated via the simulation of the Newtonian forces thereupon, falls below a predetermined height.

21. The program of any of the preceding claims, further configured to cause the processor to generate an image of a biped.

22. A computer system comprising a processor arranged to execute the program of any of the preceding claims.

23. A computer readable storage medium upon which is stored the computer program of any one of claims 1 to 21.

24. A method of controlling the locomotion of a computer-generated multiped (10), comprising the steps of:

(a) establishing a neural network having a plurality of neurones;
(b) generating a computer image of the multiped, the multiped having a body trunk (20) and at least two legs (25), each leg depending from the trunk, each leg including at least a first, upper part (35) and a second, relatively lower part (45);
(c) simulating the Newtonian forces that would be acting upon each separate part of the computer-generated multiped at a given time;
(d) applying those simulated forces to each said separate part so that, when displayed on a visual display unit, each part appears to move under the influence of those simulated forces to cause the image of each upper leg

part to pivot about the trunk and the image of each relatively lower leg part to describe an apparently substantially free-swinging movement relative to its respective upper leg part;
(e) simulating an actuator for actuation of each said leg; and
(f) controlling the operation of the actuator via an output from the neural network.

25. The method of claim 24, in which the step (a) of establishing a neural network further comprises establishing a neural network having a plurality of neurones arranged to act as two bi-directionally coupled pairs of identical oscillators.

26. The method of claim 24 or claim 25, in which the step (a) of establishing a neural network further comprises establishing fully recurrent neural networks (RNNs), having a plurality of nodes, for each of the plurality of oscillators.

27. The method of claim 26, in which the step (a) comprises establishing RNNs each comprising six nodes.

28. The method of claim 26 or claim 27, further comprising connecting each RNN in one of the chains to a laterally adjacent RNN in the other of the chains via symmetrical weights so as to provide bi-directional coupling therebetween.

29. The method of claim 28, further comprising connecting each neurone within each RNN in one of the chains asymmetrically (uni-directionally) with a corresponding neurone within a next RNN in the same chain.

30. The method of any of claims 24 to 29, further comprising:

evolving the properties of the neural network via a genetic algorithm.

31. The method of claim 30, in which the neural network properties are evolved by:

(i) generating a plurality of different trial arrays of network property values;
(ii) applying the different trial arrays to the neural network to create different trial neural networks;
(iii) identifying those arrays selected from the plurality of different arrays which, when applied to the neural networks, form trial neural networks which optimize the control of the said image of the multiped in accordance with at least the criterion that the distance, d, over which an end of the lower part of a rearmost leg of the computer-generated image should have travelled from an origin should be maximized.

32. The method of claim 31, further comprising:

aborting, as non-optimized, the consideration of those arrays forming trial neural networks once a point attractor has been reached, such that the image when controlled by that trial network stops moving away from the origin.

33. The method of claim 31 or claim 32, further comprising aborting, as non-optimized, the consideration of those arrays forming trial neural networks once the centre of mass of the multiped thus controlled drops below a predetermined threshold height.

34. The method of claim 31, claim 32 or claim 33, further comprising aborting, as non-optimized, the consideration of those arrays forming trial neural networks if the thus controlled multiped does not move relative to the origin.

35. The method of any one of claims 31 to 34, further comprising:

ranking each trial neural networks in order of fitness;
discarding those arrays representing the less fit trial neural networks;
generating further trial neural networks on the basis of the remaining trial arrays; and mutating the resulting trial arrays so as to generate a further mutated set of trial arrays;
generating further trial neural networks on the basis of the mutated trial arrays so as to allow identification of still further optimized trial arrays.

36. The method of claim 35, further comprising:

applying an ancillary stability controller to the computer-generated biped so as to stabilize sagittal and/or lateral stability during initial identification of arrays; and subsequently reducing the effect of the ancillary controller prior to identification of the said still further optimized trial arrays.

Patentansprüche

1. Computerprogramm, umfassend Computerprogrammelemente, welche, wenn sie auf einem digitalen Computer ausgeführt werden, Prozessierungsmittel dazu veranlassen:

   (i) ein Bild eines Mehrfüßlers (10), der einen Rumpf (20) und wenigstens zwei Beine (25), die von diesem abhängen, aufweist, wobei jedes Bein wenigstens einen ersten, oberen Teil (35) und einen zweiten, verhältnismäßig niedrigeren Teil (45) umfasst, auf einer visuellen Anzeigeeinheit zu erzeugen;
   (ii) die Newton'schen Kräfte, die auf jeden einzelnen Teil des Mehrfüßlers zu einer gegebenen Zeit wirken würden, zu simulieren, und diese simulierten Kräfte auf jeden der einzelnen Teile derart anzuwenden, dass, wenn er dargestellt wird, sich jeder Teil unter dem Einfluss dieser simulierten Kräfte bewegt;
   (iii) das Bild jedes oberen Beinteils zum Schwenken um den Rumpf zu veranlassen;
   (iv) das Bild des oberen Teils jedes Beins zum Bewegen mittels eines simulierten Aktuators in einem Bogen relativ zu dem Rumpf zu veranlassen;
   (v) das Bild jedes verhältnismäßig niedrigeren Beinteils dazu zu veranlassen, eine scheinbar im Wesentlichen freischwingende Bewegung relativ zu seinem jeweiligen oberen Teil zu beschreiben; und
   (vi) ein neuronales Netzwerk bereitzustellen, das eine Mehrzahl von Neuronen aufweist, wobei die Ausgabe des neuronalen Netzwerks die Aktuatorbewegung von jedem oberen Beinteil so steuert/regelt, dass das Bild des Mehrfüßlers zu laufen scheint.

2. Programm nach Anspruch 1, ferner dazu eingerichtet, die Prozessierungsmittel dazu zu veranlassen, den Aktuator durch Berechnen des simulierten Drehmoments Γ_T zu einer Zeit T, die benötigt wird, das Bild des oberen Teils jedes Beins unter der Wirkung der simulierten Newton'schen Kräfte zu bewegen, als Antwort auf eine Ausgabe des neuronalen Netzwerks zu simulieren.

3. Programm nach Anspruch 2, in welchem der Aktuator ein Proportionalableitungsaktuator ist, wobei das simulierte Drehmoment Γ_T, das auf den oberen Teil jedes Beins angewendet wird, eine Funktion des Winkels Θ des oberen Teils jedes der Beine relativ zu einem anderen Körperteil, ist.

4. Programm nach Anspruch 2, in welchem der Aktuator ein Proportional-Integral-Ableitungsaktuator ist, wobei das simulierte Drehmoment Γ_T, das auf den oberen Teil jedes Beins angewendet wird, eine Funktion des Winkels Θ des oberen Teils jedes der Beine relativ zu einem anderen Körperteil, ist.

5. Programm nach Anspruch 4, in welchem das Drehmoment Γ_T zur Zeit T mit dem Winkel Θ in Übereinstimmung mit der Gleichung

$$\Gamma_T = k_p(\theta_d - \theta) - k_D \frac{d\theta}{dt} + k_I \int_{T-\Delta T}^{T+\Delta T} (\Theta_d - \Theta)\, dt$$

   in Verbindung steht, wobei k_p, k_D und k_I die Proportional-, Ableitungs- und Integralkoeffizienten sind, und t die Zeit ist.

6. Programm nach Anspruch 1, ferner dazu eingerichtet, die Prozessierungsmittel dazu zu veranlassen, den Aktuator als einen Kraftbegrenzungs-Geschwindigkeitsbeschränkungs-Aktuator zu simulieren, in welchem das Drehmoment Γ_T, das auf den oberen Teil von jedem der zwei Beine angewendet wird, eine Funktion der SollBewegungsgeschwindigkeit des oberen Teils eines Beins relativ zum Körperrumpf ist, und wobei die Sollbewegungsgeschwindigkeit mit einer Differenz zwischen dem Ist- und dem Soll-Winkel zwischen jedem oberen Beinteil und dem Körperrumpf in Verbindung steht, wobei der Soll-Relativwinkel von dem neuronalen Netzwerk abgeleitet wird.

7. Programm nach einem der vorhergehenden Ansprüche, ferner dazu eingerichtet, die Prozessierungsmittel dazu zu veranlassen, ein simuliertes Hüftgelenk für jedes Bein zu erzeugen, um den oberen Teil eines bestimmten Beins an den Rumpf zu verbinden, und wenigstens einen Aktuator pro Bein zu simulieren, wobei jeder Aktuator dazu eingerichtet ist, auf das damit verbundene Hüftgelenk so zu wirken, dass das Bild des zugehörigen unteren Beinteils

zum Bewegen veranlasst wird.

8. Programm nach Anspruch 7, ferner dazu eingerichtet, die Prozessierungsmittel zum Simulieren eines Aktuators pro Bein zu veranlassen.

9. Programm nach Anspruch 7 oder Anspruch 8, ferner dazu eingerichtet, den Prozessor zum Einschränken des simulierten Hüftgelenks, darauf nur einen Freiheitsgrad zu besitzen, zu veranlassen, so dass das Bild des zugehörigen oberen Beinteils darauf eingeschränkt ist, sich nur sagittal zu bewegen.

10. Programm nach einem der vorhergehenden Ansprüche, ferner dazu eingerichtet, die Prozessierungsmittel zum Erzeugen eines Bildes einer Mehrzahl von Füßen zu veranlassen, von welchem jeder an ein Ende des Bildes des unteren Teils eines zugehörigen Beins mittels eines Knöchelgelenks verbunden zu sein scheint, und wobei die Prozessierungsmittel die Wirkung einer gedämpften Torsionsfeder zum Regulieren der Bewegung des Knöchelgelenks simulieren.

11. Programm nach einem der vorhergehenden Ansprüche, ferner dazu eingerichtet, die Prozessierungsmittel zum Begrenzen des Winkels zwischen dem Bild des oberen Teils und dem Bild des unteren Teils von jedem Bein zu veranlassen.

12. Programm nach einem der vorhergehenden Ansprüche, in welchem die Neuronen des neuronalen Netzwerks als rekurrentes neuronales Netzwerk (RNN) derart eingerichtet sind, dass sie zwei bidirektional gekoppelte Ketten von identischen Oszillatoren bilden, wobei wenigstens zwei der Neuronen als Motoneuronen agieren, deren Ausgabe den Aktuator, der dem oberen Teil des Beins zugehörig ist, steuert/regelt.

13. Programm nach Anspruch 12, in welchem das RNN eine Mehrzahl von Neuronen umfasst, die so eingerichtet sind, dass sie bidirektional gekoppelte Paare von identischen Oszillatoren bilden.

14. Programm nach Anspruch 13, in welchem das RNN ein rekurrentes Subnetz pro Oszillator umfasst, von denen jedes sechs Knoten aufweist.

15. Programm nach Anspruch 14, in welchem jedes rekurrentes Subnetz in einer der Ketten mit einem lateral angrenzenden Subnetz in der anderen der Ketten mittels symmetrischer Gewichte derart verbunden ist, dass bidirektionale Kopplung dazwischen hergestellt wird, und in welchem jede Neurone in jedem rekurrenten Subnetz zum Synapsen mit einer äquivalenten Neurone in dem lateral angrenzenden rekurrenten Subnetz eingerichtet ist.

16. Programm nach Anspruch 15, in welchem jede Neurone innerhalb jedes rekurrenten Subnetzes in einer der Ketten asymmetrisch (unidirektional) mit einer zugehörigen Neurone innerhalb eines nächsten rekurrenten Subnetzes in derselben Kette verbunden ist.

17. Programm nach einem der vorhergehenden Ansprüche, ferner dazu konfiguriert, die Prozessierungsmittel dazu zu veranlassen, die Eigenschaften der Neuronen und ihrer Verbindungen innerhalb des neuronalen Netzwerks auf der Grundlage eines genetischen Algorithmus zu bestimmen.

18. Programm nach Anspruch 17, ferner dazu konfiguriert, die Prozessierungsmittel dazu zu veranlassen, die Eigenschaften auf der Grundlage eines genetischen Algorithmus zu bestimmen, in welchem die Eigenschaften optimiert sind durch die Identifizierung eines geeignetsten Feldes von Eigenschaftswerten aus einer Mehrzahl von verschiedenen Feldern von Eigenschaftswerten in Übereinstimmung mit dem Kriterium, dass die Strecke, über welche ein Ende des unteren Teils des hintersten Beins in dem Bild des Mehrfüßlers, relativ zu einem Ursprung, eine maximale Strecke von dem Ursprung zurückgelegt haben sollte.

19. Programm nach Anspruch 18, ferner dazu konfiguriert, die Prozessierungsmittel zum Abbrechen des In-Betracht-Ziehens eines gegebenen aus den Feldern von Eigenschaftswerten zu veranlassen, wenn bestimmt wird, dass ein Punktattraktor derart erreicht wurde, dass das Bild des Mehrfüßlers aufhört, sich vom Ursprung wegzubewegen.

20. Programm nach Anspruch 18 oder Anspruch 19, ferner dazu konfiguriert, die Prozessierungsmittel zum Abbrechen des In-Betracht-Ziehens eines bestimmten der Felder zu veranlassen, wenn bestimmt wird, dass der Schwerpunkt des Rumpfs des Mehrfüßlers, der mittels der Simulation der Newton'schen Kräfte auf denselben berechnet wird, unter eine vorbestimmte Höhe fällt.

21. Programm nach einem der vorhergehenden Ansprüche, ferner dazu konfiguriert, den Prozessor zum Erzeugen eines Bildes eines Zweifüßlers zu veranlassen.

22. Computersystem umfassend einen Prozessor, der dazu eingerichtet ist, das Programm nach einem der vorhergehenden Ansprüche auszuführen.

23. Computerlesbares Speichermedium, auf welchem das Computerprogramm nach einem der Ansprüche 1 bis 21 gespeichert ist.

24. Verfahren zum Steuern/Regeln der Fortbewegung eines computer-erzeugten Mehrfüßlers (10), umfassend die Schritte:

(a) Aufbauen eines neuronalen Netzwerks umfassend eine Mehrzahl von Neuronen;
(b) Erzeugen eines Computerbildes des Mehrfüßlers, wobei der Mehrfüßler einen Körperrumpf (20) und wenigstens zwei Beine (25) aufweist, wobei jedes Bein von dem Rumpf abhängt, wobei jedes Bein wenigstens einen ersten, oberen Teil (35) und einen zweiten, verhältnismäßig niedrigeren Teil (45) umfasst;
(c) Simulieren der Newton'schen Kräfte, die auf jeden einzelnen Teil des computer-erzeugten Mehrfüßlers zu einer gegebenen Zeit wirken;
(d) Anwenden dieser simulierten Kräfte auf jeden einzelnen Teil derart dass, wenn er auf einer visuellen Anzeigeeinheit dargestellt wird, sich jeder Teil unter dem Einfluss dieser simulierten Kräfte zu bewegen scheint, um das Bild jedes oberen Beinteils zum Schwenken um den Rumpf und das Bild jedes verhältnismäßig niedrigeren Beinteils eine scheinbar im Wesentlichen freischwingende Bewegung relativ zu seinem jeweiligen oberen Beinteil zu beschreiben, zu veranlassen.
(e) Simulieren eines Aktuators zur Betätigung jedes der Beine; und
(f) Steuern/Regeln des Betriebs des Aktuators mittels einer Ausgabe von dem neuronalen Netzwerk.

25. Verfahren nach Anspruch 24, in welchem der Schritt (a) des Aufbauens eines neuronalen Netzwerks ferner das Aufbauen eines neuronalen Netzwerks umfassend eine Mehrzahl von Neuronen, die dazu eingerichtet sind, als bidirektional gekoppelte Paare von identischen Oszillatoren zu agieren, umfasst.

26. Verfahren nach Anspruch 24 oder Anspruch 25, in welchem der Schritt (a) des Aufbauens eines neuronalen Netzwerks ferner Aufbauen vollständig rekurrenter neuronaler Netzwerke (RNNs) umfasst, die eine Mehrzahl von Knoten für jede der Mehrzahl von Oszillatoren, aufweisen.

27. Verfahren nach Anspruch 26, in welchem der Schritt (a) Aufbauen von RNNs, jeweils umfassend sechs Knoten, umfasst.

28. Verfahren nach Anspruch 26 oder Anspruch 27, ferner umfassend Verbinden jedes RNN in einer der Ketten mit einem lateral angrenzenden RNN in der anderen der Ketten mittels symmetrischer Gewichte, um bidirektionale Kopplung zwischen jenen bereitzustellen.

29. Verfahren nach Anspruch 28, ferner umfassend asymmetrisches (unidirektionales) Verbinden jeder Neurone innerhalb jedes RNN in einer der Ketten mit einer entsprechenden Neurone innerhalb eines nächsten RNN in derselben Kette.

30. Verfahren nach einem der Ansprüche 24 bis 29, ferner umfassend:

Entwickeln der Eigenschaften des neuronalen Netzwerks mittels eines genetischen Algorithmus.

31. Verfahren nach Anspruch 30, in welchem die Eigenschaften des neuronalen Netzwerks entwickelt werden durch:

(i) Erzeugen einer Mehrzahl von verschiedenen Versuchsfeldern von Netzwerkeigenschaftswerten;
(ii) Anwenden der verschiedenen Versuchsfelder auf das neuronale Netzwerk zum Erzeugen von verschiedenen Versuchs-Neuronalen-Netzwerken;
(iii) Identifizieren derjenigen Felder, die aus der Mehrzahl von verschiedenen Feldern ausgewählt wurden, die, wenn sie auf die neuronalen Netzwerke angewendet werden, Versuchs-Neuronale-Netzwerke bilden, die die Steuerung/Regelung des Bildes des Mehrfüßlers in Übereinstimmung mit wenigstens dem Kriterium, dass die Strecke d, über welche sich ein Ende des unteren Teils des hintersten Beins des computer-erzeugten Bildes

von einem Ursprung bewegt haben sollte, maximiert werden sollte, optimieren.

32. Verfahren nach Anspruch 31, ferner umfassend:

Abbrechen des In-Betracht-Ziehens, als nicht optimiert, derjenigen Felder, die Versuchs-Neuronale-Netzwerke bilden, sobald ein Punktattraktor erreicht worden ist, derart dass das Bild, wenn es von diesem Versuchs-Netzwerk gesteuert/geregelt wird, aufhört, sich vom Ursprung wegzubewegen.

33. Verfahren nach Anspruch 31 oder Anspruch 32, ferner umfassend Abbrechen des In-Betracht-Ziehens, als nicht optimiert, derjenigen Felder, die Versuchs-Neuronale-Netzwerke bilden, sobald der Schwerpunkt des Mehrfüßlers, der dadurch gesteuert/geregelt wird, unter eine vorbestimmte Schwellenhöhe fällt.

34. Verfahren nach Anspruch 31, Anspruch 32 oder Anspruch 33, ferner umfassend Abbrechen des In-Betracht-Ziehens, als nicht optimiert, derjenigen Felder, die Versuchs-Neuronale-Netzwerke bilden, wenn der dadurch gesteuerte/geregelte Mehrfüßler sich nicht relativ zum Ursprung bewegt.

35. Verfahren nach einem der Ansprüche 31 bis 34, ferner umfassend:

Einordnen nach Rängen jedes Versuchs-Neuronalen-Netzwerks in Reihenfolge seiner Eignung;
Verwerfen derjenigen Felder, die die weniger geeigneten Versuchs-Neuronalen-Netzwerke repräsentieren;
Erzeugen weiterer Versuchs-Neuronaler-Netzwerke auf der Grundlage der verbleibenden Versuchsfelder; und
Abwandeln der resultierenden Versuchsfelder derart, dass eine weitere abgewandelte Menge von Versuchsfeldern erzeugt wird;
Erzeugen weiterer Versuchs-Neuronaler-Netzwerke auf der Grundlage der abgewandelten Versuchsfelder, so dass eine Identifikation von noch weiter optimierten Versuchsfeldern erlaubt ist.

36. Verfahren nach Anspruch 35, ferner umfassend:

Anwenden einer Hilfsstabilitätssteuerung/-regelung auf den computer-erzeugten Zweifüßler, so dass die sagittale und/oder laterale Stabilität während der anfänglichen Identifizierung der Felder stabilisiert wird; und anschließend Reduzieren des Effekts der Hilfssteuerung/- regelung vor Identifizierung der noch weiter optimierten Versuchsfelder.

Revendications

1. Programme informatique comprenant des éléments de programme informatique qui, lorsqu'ils sont exécutés sur un ordinateur numérique, amènent un moyen de traitement correspondant :

(i) à générer une image, sur une unité de visualisation, d'un multipède (10) ayant un tronc (20) et au moins deux jambes (25) qui en dépendent, chaque jambe comportant au moins une première partie supérieure (35) et une deuxième partie relativement inférieure (45) ;
(ii) à simuler les forces newtoniennes qui seraient amenées à agir sur chaque partie séparée du multipède à un instant donné et à appliquer ces forces simulées sur chacune desdites parties séparées de sorte que, lorsqu'elles sont affichées, chaque partie se déplace sous l'influence de ces forces simulées ;
(iii) à provoquer le pivotement de l'image de chaque partie supérieure de jambe autour du tronc ;
(iv) à provoquer le déplacement de l'image de la partie supérieure de chaque jambe, par l'intermédiaire d'un actionneur simulé, dans un arc par rapport audit tronc ;
(v) à amener l'image de chaque partie relativement inférieure de jambe à décrire un mouvement d'oscillation paraissant sensiblement libre par rapport à sa partie supérieure respective ; et
(vi) à fournir un réseau de neurones comportant une pluralité de neurones, la sortie du réseau de neurones commande le mouvement de l'actionneur de chaque partie supérieure de jambe de sorte que l'image du multipède semble marcher.

2. Programme de la revendication 1, agencé en outre pour amener le moyen de traitement à simuler l'actionneur en calculant le couple simulé, Γ_T, à un instant T qui est nécessaire pour déplacer l'image de la partie supérieure de chaque jambe, sous l'effet desdites forces newtoniennes simulées, en réponse à une sortie provenant du réseau de neurones.

3. Programme de la revendication 2, dans lequel l'actionneur est un actionneur à action proportionnelle-dérivée, où le couple simulé, Γ_T, appliqué à la partie supérieure de chaque jambe dépend de l'angle, θ, de la partie supérieure de chacune desdites jambes par rapport à une autre partie du corps.

4. Programme de la revendication 2, dans lequel l'actionneur est un actionneur à action proportionnelle-intégrale-dérivée où le couple simulé, Γ_T, appliqué à la partie supérieure de chaque jambe dépend de l'angle, θ, de la partie supérieure de chacune desdites jambes par rapport à une autre partie du corps.

5. Programme de la revendication 4, dans lequel le couple Γ_T, à l'instant T est lié à l'angle θ conformément à l'équation

$$\Gamma_T = k_p(\theta_d - \theta) - k_D \frac{d\theta}{dt} + k_I \int_{T-\Delta T}^{T+\Delta T} (\theta_d - \theta) dt$$

où k_p, k_D et k_I sont les coefficients de proportionnalité, de dérivation et d'intégration, et t est le temps.

6. Programme de la revendication 1, agencé en outre pour amener le moyen de traitement à simuler l'actionneur comme étant un actionneur à contrainte de vitesse à force limitée où le couple, Γ_T, appliqué à la partie supérieure de chacune des deux jambes dépend de la vitesse souhaitée de déplacement de la partie supérieure d'une jambe par rapport au tronc du corps, et où la vitesse souhaitée de déplacement est liée à une différence entre l'angle réel et l'angle souhaité entre chaque partie supérieure de jambe et le tronc du corps, l'angle relatif souhaité étant dérivé du réseau de neurones.

7. Programme de l'une quelconque des revendications précédentes, agencé en outre pour amener le moyen de traitement à générer une articulation de hanche simulée pour chaque jambe, pour lier la partie supérieure d'une jambe donnée au tronc et pour simuler au moins un actionneur par jambe, chaque actionneur étant agencé pour agir sur l'articulation de hanche qui lui est associée afin d'amener l'image de la partie supérieure de jambe correspondante à se déplacer.

8. Programme de la revendication 7, agencé en outre pour amener le moyen de traitement à simuler un actionneur par jambe.

9. Programme de la revendication 7 ou 8, agencé en outre pour amener le processeur à contraindre l'articulation de hanche simulée à avoir un seul degré de liberté de sorte que l'image de la partie supérieure de jambe correspondante soit contrainte à ne se déplacer que sagittalement.

10. Programme de l'une quelconque des revendications précédentes, agencé en outre pour amener le moyen de traitement à générer une image d'une pluralité de pieds, dont chacune semble être reliée à une extrémité de l'image de la partie inférieure d'une jambe correspondante par l'intermédiaire d'une articulation de cheville, et où le moyen de traitement simule l'action d'un ressort de torsion amorti pour diriger le déplacement de chacune desdites articulations de cheville.

11. Programme de l'une quelconque des revendications précédentes, agencé en outre pour amener le moyen de traitement à limiter l'angle entre l'image de la partie supérieure et l'image de la partie inférieure de chaque jambe.

12. Programme de l'une des revendications précédentes, dans lequel les neurones du réseau de neurones sont agencés sous forme d'un réseau de neurones récurrent (RNR) de façon à former deux chaînes d'oscillateurs identiques couplées de manière bidirectionnelle, au moins deux des neurones agissant comme neurones moteurs dont la sortie commande ledit actionneur associé à la partie supérieure de chacune desdites jambes.

13. Programme de la revendication 12, dans lequel le réseau RNR comprend une pluralité de neurones agencés de manière à former deux paires d'oscillateurs identiques couplées de manière bidirectionnelle.

14. Programme de la revendication 13, dans lequel le réseau RNR comprend un sous-réseau récurrent par oscillateur, dont chacun comporte six noeuds.

15. Programme de la revendication 14, dans lequel chaque sous-réseau récurrent dans l'une des chaînes est relié à un sous-réseau récurrent latéralement adjacent dans l'autre des chaînes par l'intermédiaire de poids symétriques de manière à fournir un accouplement bidirectionnel entre eux, et dans lequel chaque neurone dans chaque sous-réseau récurrent est agencé pour établir une synapse avec un neurone équivalent dans ledit sous-réseau récurrent latéralement adjacent.

16. Programme de la revendication 15, dans lequel chaque neurone dans chaque sous-réseau récurrent dans l'une des chaînes latérales est relié de manière asymétrique (unidirectionnelle) à un neurone correspondant dans un sous-réseau récurrent suivant dans la même chaîne.

17. Programme de l'une quelconque des revendications précédentes, configuré en outre pour amener le moyen de traitement à déterminer les propriétés des neurones et de leurs connexions dans ledit réseau de neurones sur la base d'un algorithme génétique.

18. Programme de la revendication 17, configuré en outre pour amener le moyen de traitement à déterminer lesdites propriétés sur la base d'un algorithme génétique dans lequel lesdites propriétés sont optimisées par l'identification de la matrice de valeurs de propriétés la plus compatible parmi une pluralité de matrices de valeurs de propriétés différentes sur la base d'un critère de maximisation de la distance que doit parcourir, par rapport à une origine, une extrémité de la partie inférieure de la jambe la plus en arrière dans l'image du multipède.

19. Programme de la revendication 18, configuré en outre pour amener le moyen de traitement à abandonner la prise en compte de l'une des matrices de valeurs de propriétés si l'on détermine qu'un point attracteur a été atteint de sorte que l'image du multipède arrête de s'éloigner de l'origine.

20. Programme de la revendication 18 ou 19, configuré en outre pour amener le moyen de traitement à abandonner la prise en compte d'une matrice particulière parmi les matrices si l'on détermine que le centre de masse du tronc du multipède, tel que calculé par la simulation des forces newtoniennes agissant dessus, est inférieur à une hauteur prédéterminée.

21. Programme de l'une des revendications précédentes, configuré en outre pour amener le processeur à générer une image d'un bipède.

22. Système informatique comprenant un processeur agencé pour exécuter le programme de l'une des revendications précédentes.

23. Support de stockage lisible par ordinateur sur lequel est stocké le programme informatique de l'une quelconque des revendications 1 à 21.

24. Procédé de commande de la locomotion d'un multipède (10) généré par ordinateur, comprenant les étapes qui consistent :

(a) à établir un réseau de neurones ayant une pluralité de neurones ;
(b) à générer une image numérique du multipède, le multipède ayant un tronc (20) de corps et au moins deux jambes (25), chaque jambe dépendant du tronc, chaque jambe comportant au moins une première partie supérieure (35) et une deuxième partie relativement inférieure (45) ;
(c) à simuler les forces newtoniennes qui seraient amenées à agir sur chaque partie séparée du multipède généré par ordinateur à un instant donné ;
(d) à appliquer ces forces simulées à chacune desdites parties séparées de sorte que, lorsqu'elles sont affichées sur une unité de visualisation, chaque partie semble se déplacer sous l'influence de ces forces pour amener l'image de chaque partie de jambe supérieure à pivoter autour du tronc et l'image de chaque partie de jambe relativement inférieure à décrire un mouvement d'oscillation paraissant sensiblement libre par rapport à sa partie de jambe supérieure respective ;
(e) à simuler un actionneur pour actionner chacune desdites jambes ; et
(f) à commander le fonctionnement de l'actionneur par l'intermédiaire d'une sortie provenant du réseau de neurones.

25. Procédé de la revendication 24, dans lequel l'étape (a) d'établissement d'un réseau de neurones comprend en outre le fait d'établir un réseau de neurones ayant une pluralité de neurones agencés pour agir comme deux paires d'oscillateurs identiques couplées de manière bidirectionnelle.

26. Procédé de la revendication 24 ou 25, dans lequel l'étape (a) d'établissement d'un réseau de neurones comprend en outre le fait d'établir des réseaux de neurones entièrement récurrents (RNNs), ayant une pluralité de noeuds, pour chacun de la pluralité d'oscillateurs.

27. Procédé de la revendication 26, dans lequel l'étape (a) comprenant le fait d'établir des réseaux RNN comprenant chacun six noeuds.

28. Procédé de la revendication 26 ou 27, comprenant en outre le fait de relier chaque réseau RNN dans l'une des chaînes à un réseau RNN latéralement adjacent dans l'autre des chaînes par l'intermédiaire de poids symétriques de manière à fournir un accouplement bidirectionnel entre eux.

29. Procédé de la revendication 28, comprenant en outre le fait de relier chaque neurone dans chaque réseau RNN dans l'une des chaînes de manière asymétrique (unidirectionnelle) avec un neurone correspondant dans un réseau RNN suivant dans la même chaîne.

30. Procédé de l'une des revendications 24 à 29, comprenant en outre le fait :

de développer les propriétés du réseau de neurones par l'intermédiaire d'un algorithme génétique.

31. Procédé de la revendication 30, dans lequel les propriétés du réseau de neurones sont développées :

(i) en générant une pluralité de matrices de valeurs de propriétés d'essai différentes du réseau ;
(ii) en appliquant les différentes matrices d'essai au réseau de neurones pour créer différents réseaux de neurones d'essai ;
(iii) en identifiant ces matrices sélectionnées parmi la pluralité de matrices différentes qui, lorsqu'elles sont appliquées aux réseaux de neurones, forment des réseaux de neurones d'essai qui optimisent la commande de ladite image du multipède sur la base d'au moins le critère de maximisation de la distance d, par laquelle une extrémité de la partie inférieure de la jambe la plus en arrière de l'image générée par ordinateur devrait se déplacer par rapport à une origine.

32. Procédé de la revendication 31, comprenant en outre le fait :

d'abandonner, les considérant comme non optimisées, la prise en compte des matrices qui forment des réseaux de neurones d'essai dès lors qu'un point attracteur a été atteint, de sorte que l'image lorsqu'elle est commandée par ce réseau d'essai arrête de s'éloigner de l'origine.

33. Procédé de la revendication 31 ou 32, comprenant en outre le fait d'abandonner, les considérant comme non optimisées, la prise en compte des matrices qui forment des réseaux de neurones d'essai dès lors que le centre de masse du multipède ainsi commandé passe sous une hauteur seuil prédéterminée.

34. Procédé de la revendication 31, 32 ou 33, comprenant en outre le fait d'abandonner, les considérant comme non optimisées, la prise en compte des matrices qui forment des réseaux de neurones d'essai si le multipède ainsi commandé ne se déplace pas par rapport à l'origine.

35. Procédé de l'une quelconque des revendications 31 à 34, comprenant en outre le fait :

de classer chaque réseau de neurones d'essai dans l'ordre de compatibilité ;
de rejeter les matrices qui représentant les réseaux de neurones d'essai les moins compatibles ;
de générer des réseaux de neurones d'essai supplémentaires sur la base des matrices d'essai restantes ; et
de muter les matrices d'essai résultantes de manière à générer un ensemble muté supplémentaire de matrices d'essai ;
de générer des réseaux de neurones d'essai supplémentaires sur la base des matrices d'essai mutées de manière à permettre l'identification d'autres réseaux d'essai optimisés supplémentaires.

36. Procédé de la revendication 35, comprenant en outre le fait :

d'appliquer une commande de stabilité auxiliaire au bipède généré par ordinateur de manière à stabiliser la stabilité latérale et/ou sagittale pendant l'identification initiale des matrices ; et de réduire par la suite l'effet de la commande auxiliaire avant l'identification desdits autres réseaux d'essai optimisés supplémentaires.

FIG. 1.

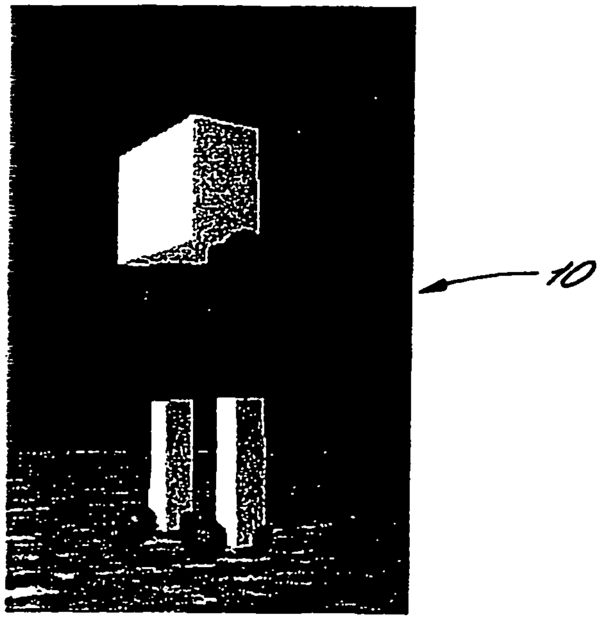


FIG. 1b.

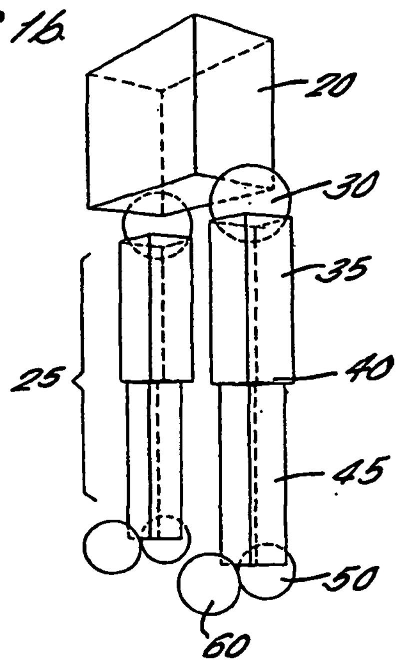


FIG. 2a.

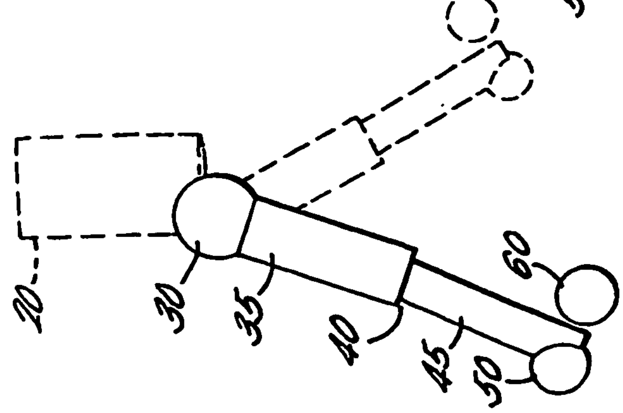


FIG. 2b.

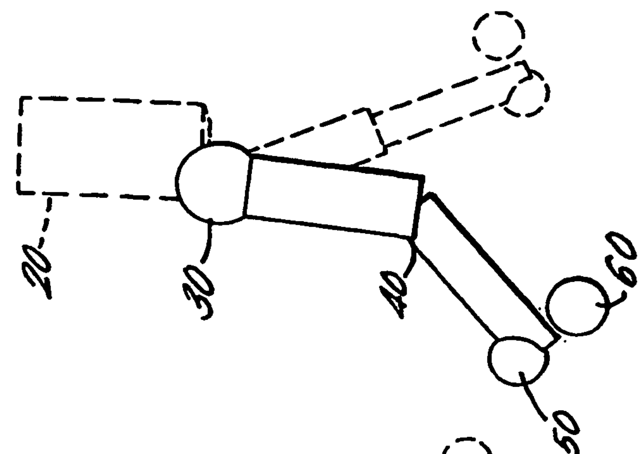


FIG. 2c.

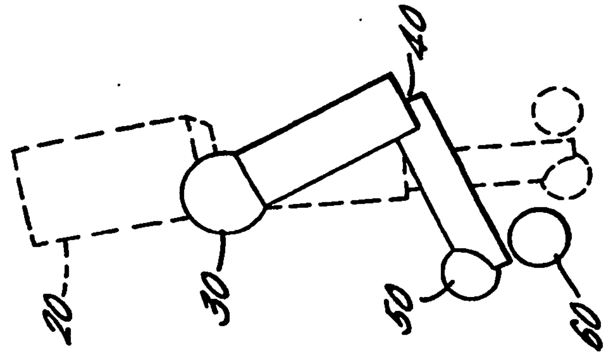


FIG. 2d.

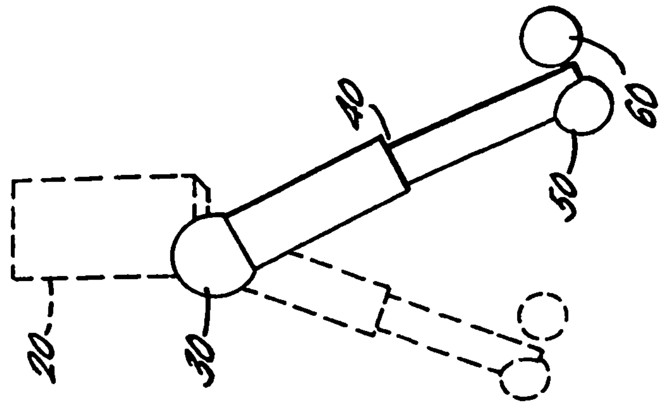
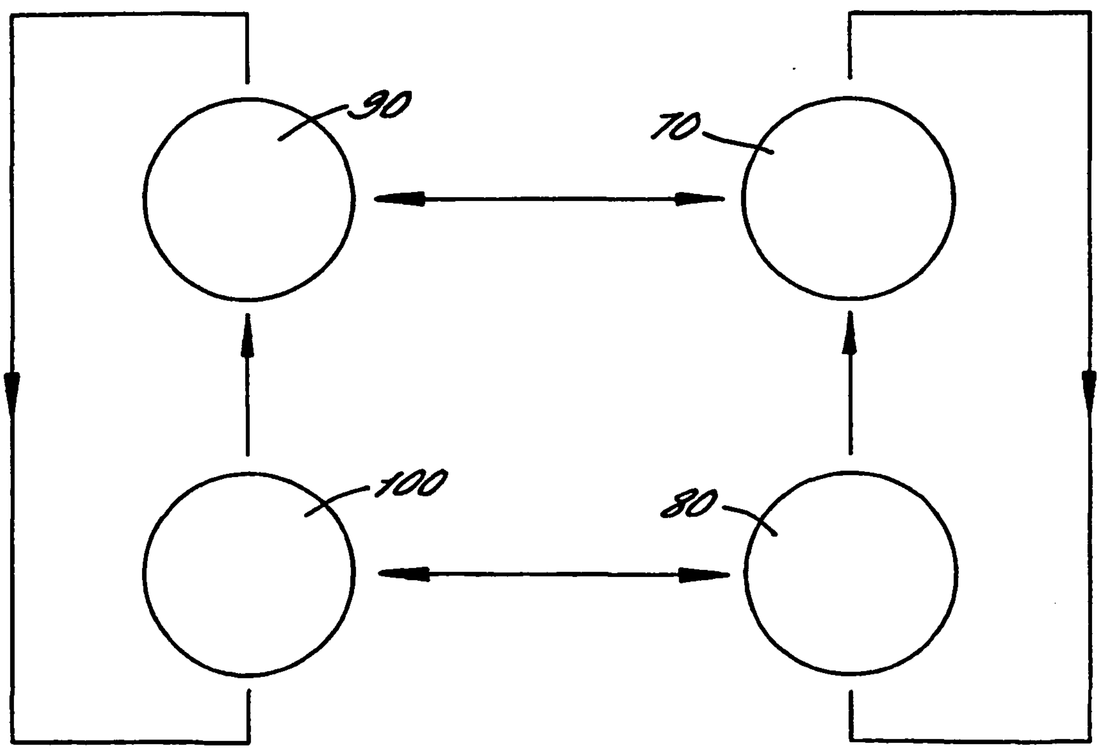


FIG. 3.



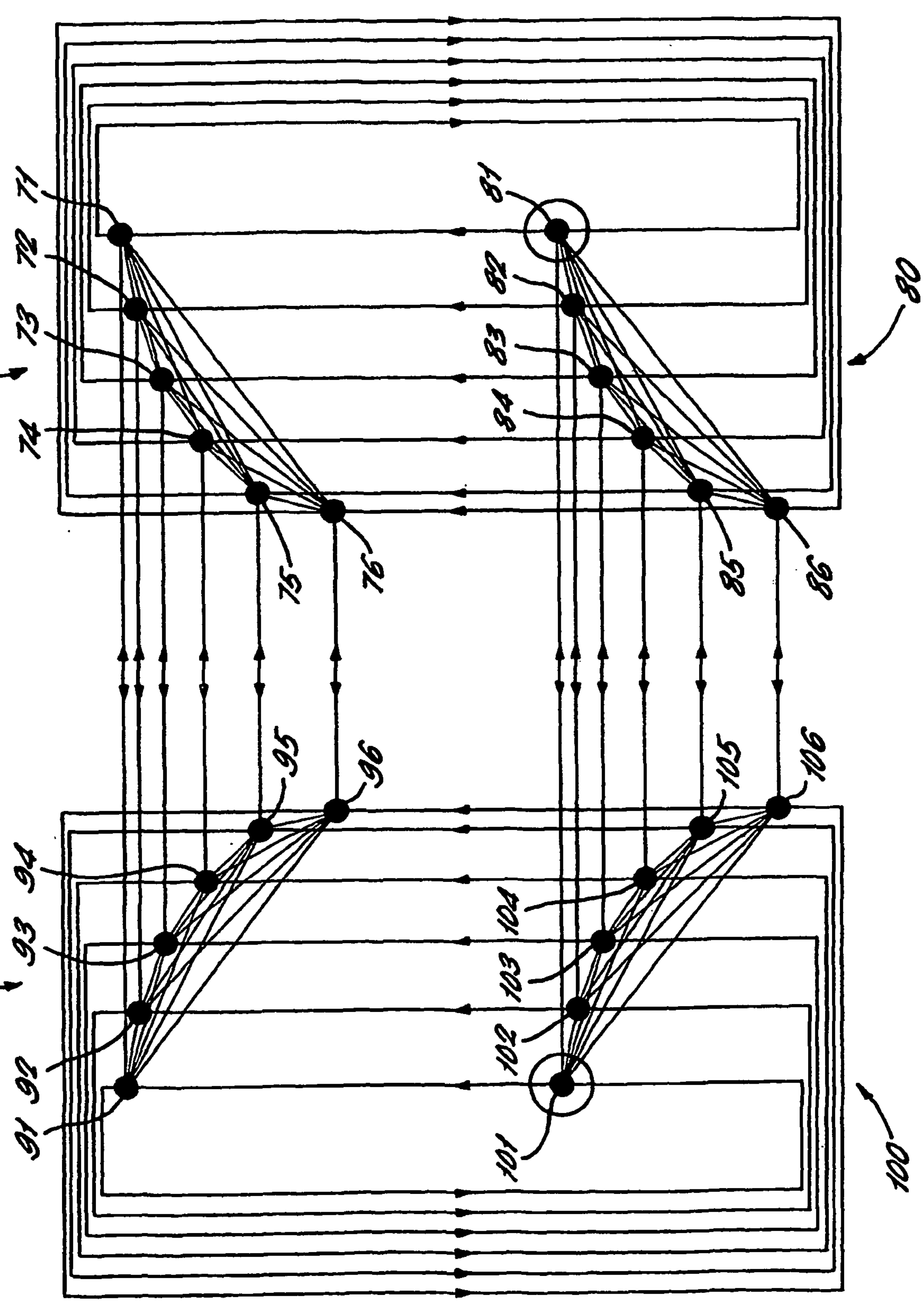

70
71
72
73
74
75
76
80
81
82
83
84
85
86
90
91
92
93
94
95
96
100
101
102
103
104
105
106

FIG.4.

FIG. 5.

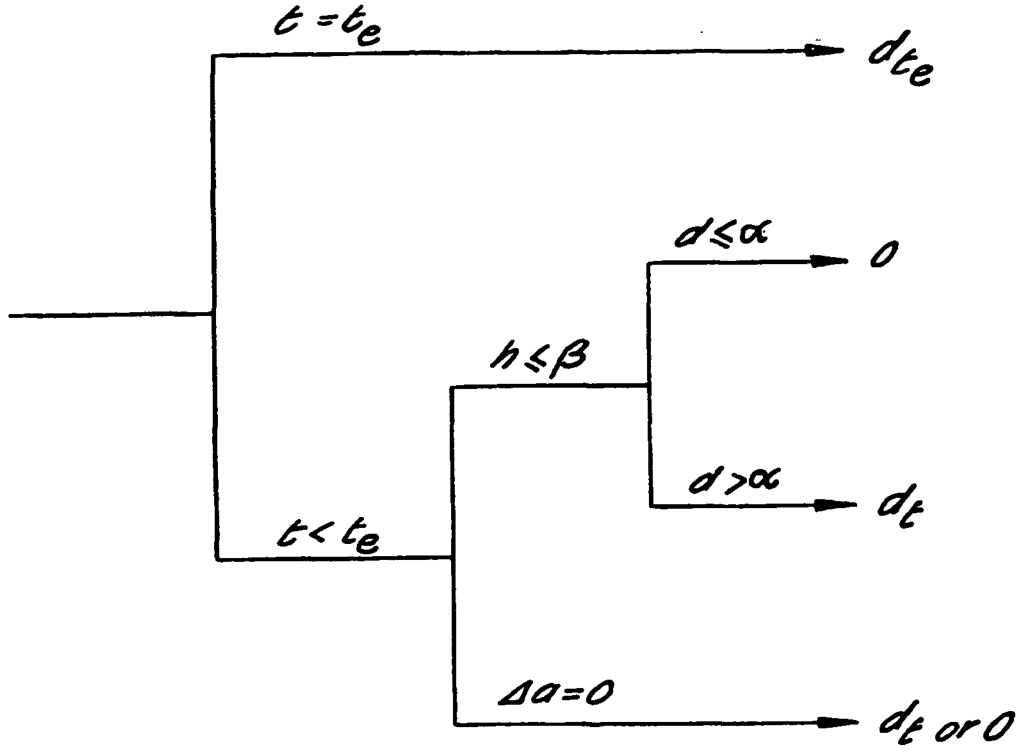


FIG. 6a.

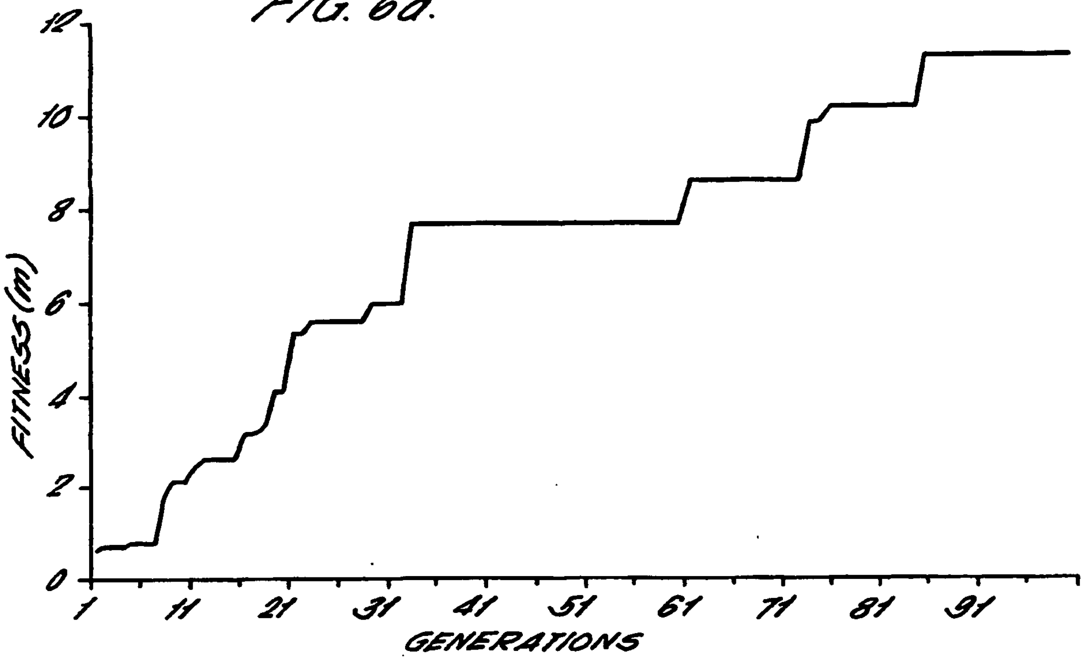

12
10
8
6
4
2
0
FITNESS (m)
1
11
21
31
41
51
61
71
81
91
GENERATIONS

FIG. 6b.

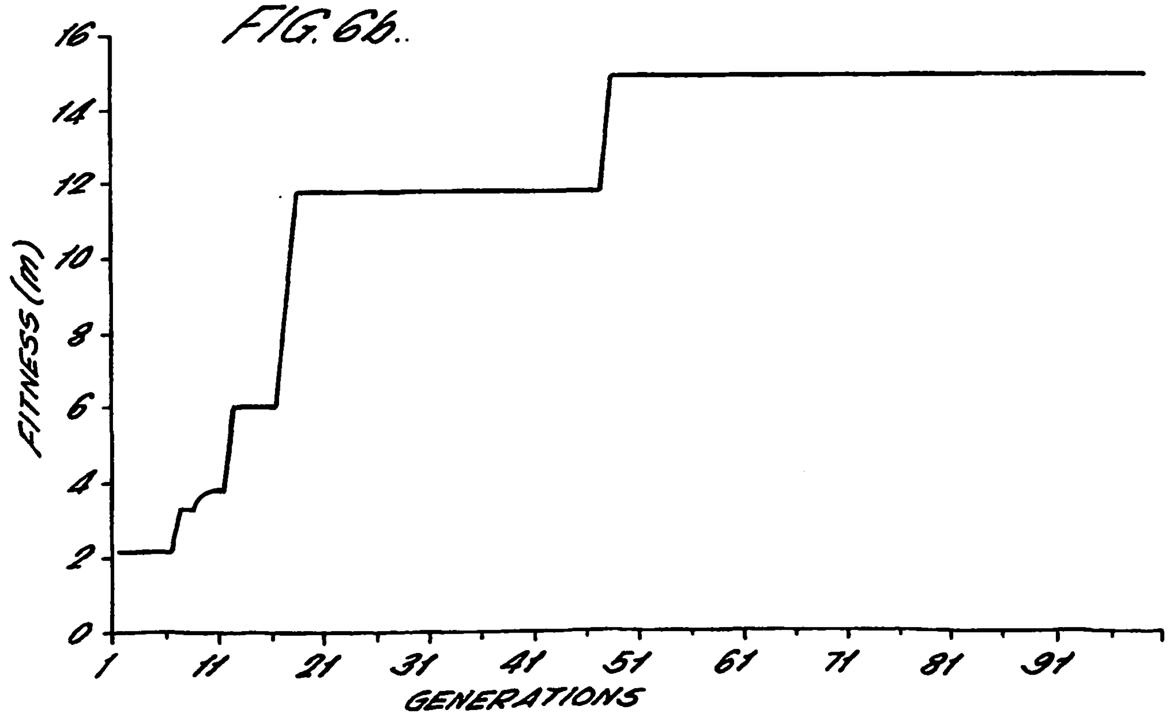

16
14
12
10
8
6
4
2
0
FITNESS (m)
1
11
21
31
41
51
61
71
81
91
GENERATIONS

FIG. 7a.

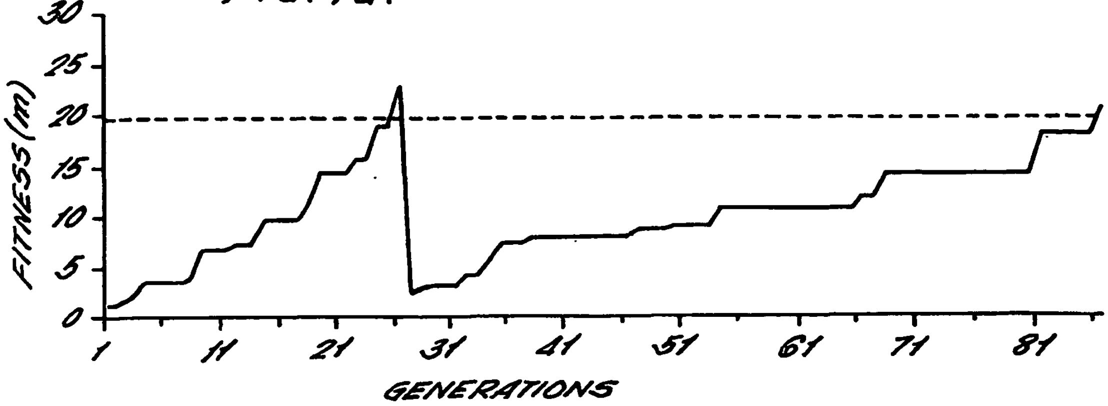


FIG. 7b.

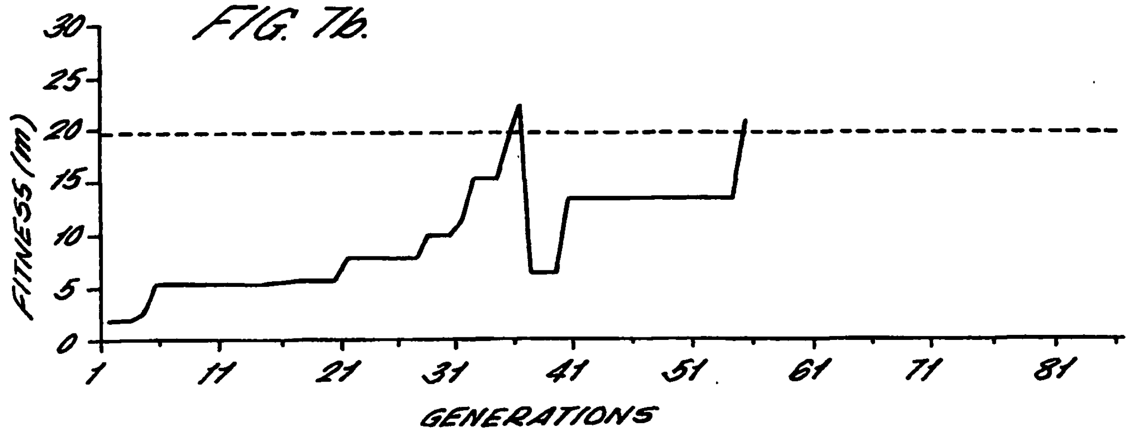
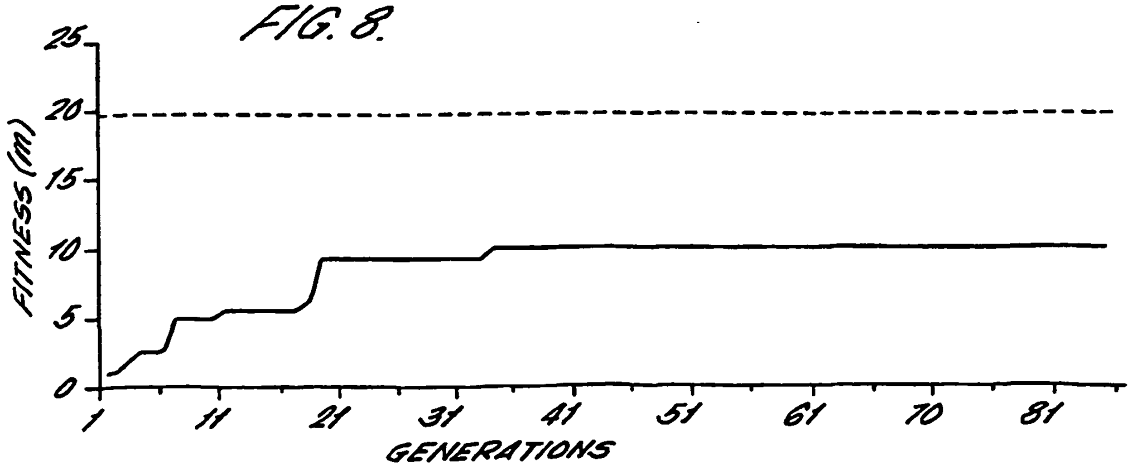


FIG. 8.



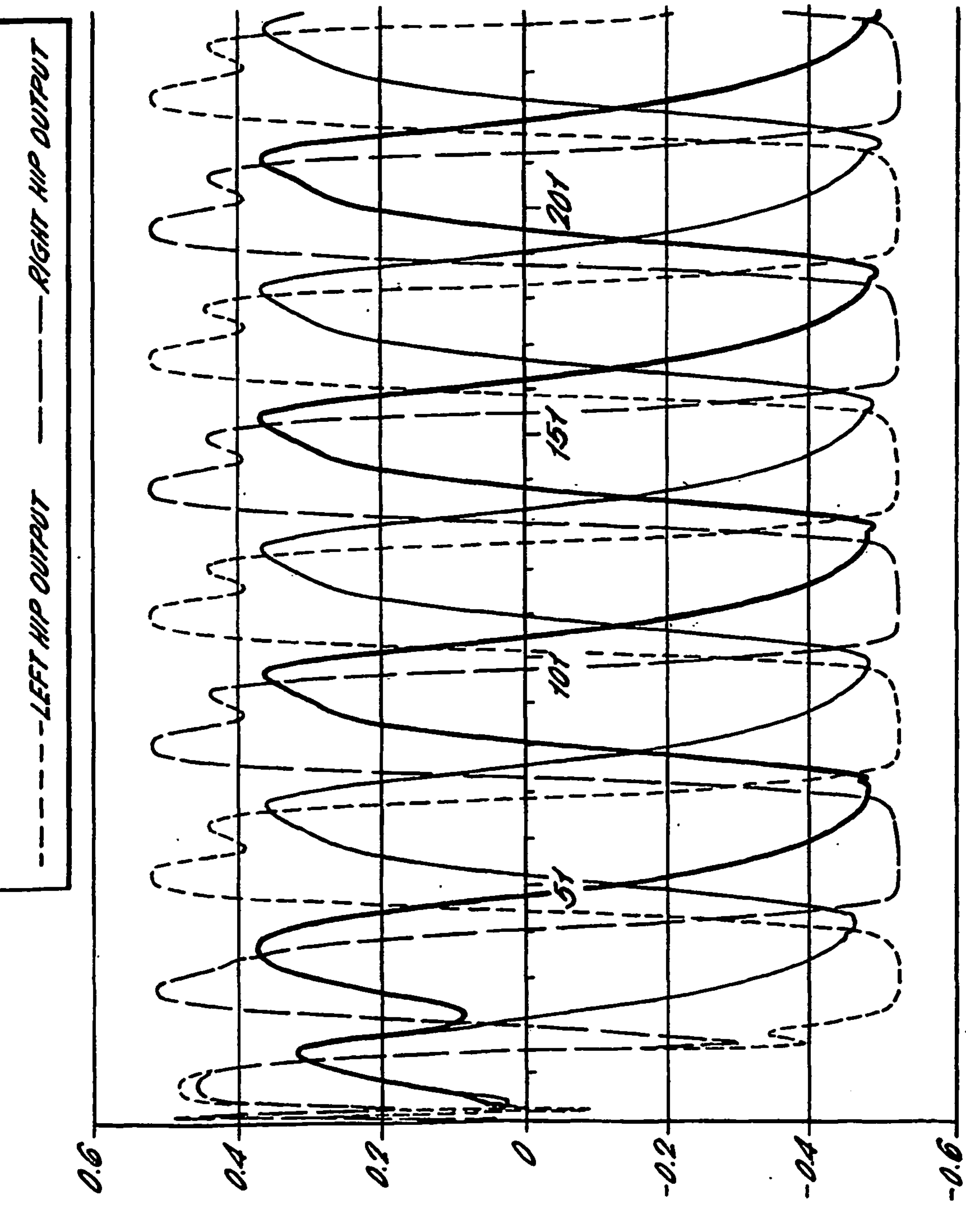

LEFT HIP ANGLE
RIGHT HIP ANGLE
LEFT HIP OUTPUT
RIGHT HIP OUTPUT
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
51
101
151
201

FIG. 9.

FIG. 10.

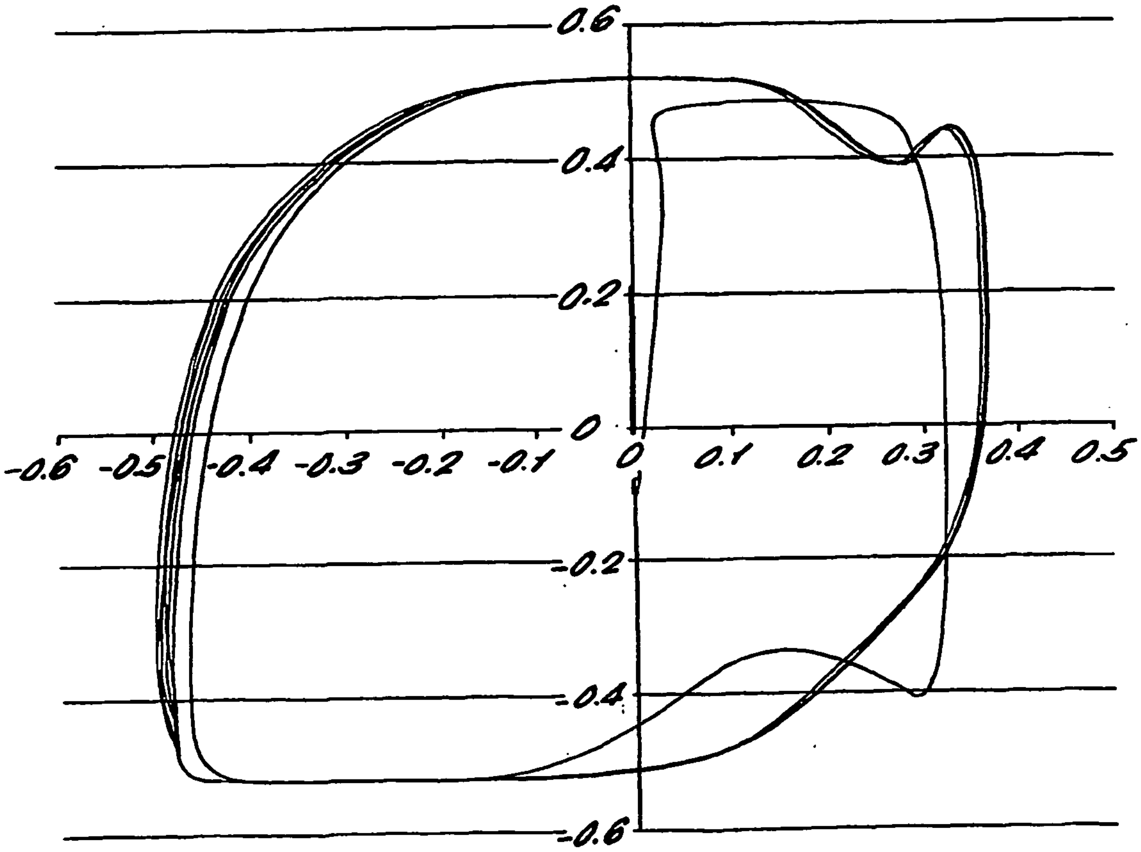


FIG. 11a.

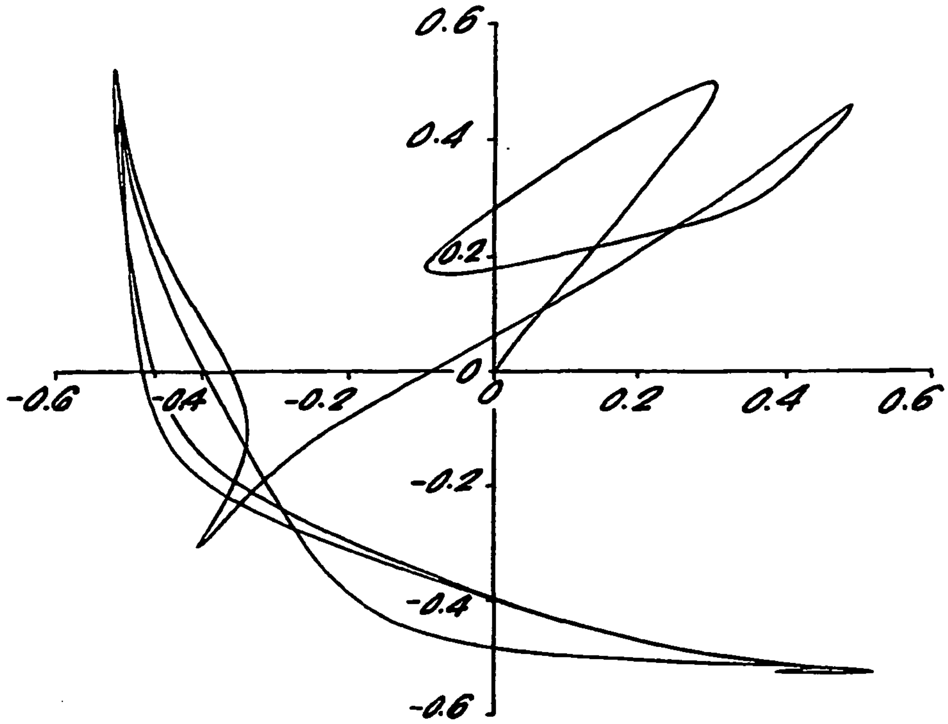


FIG. 11b.

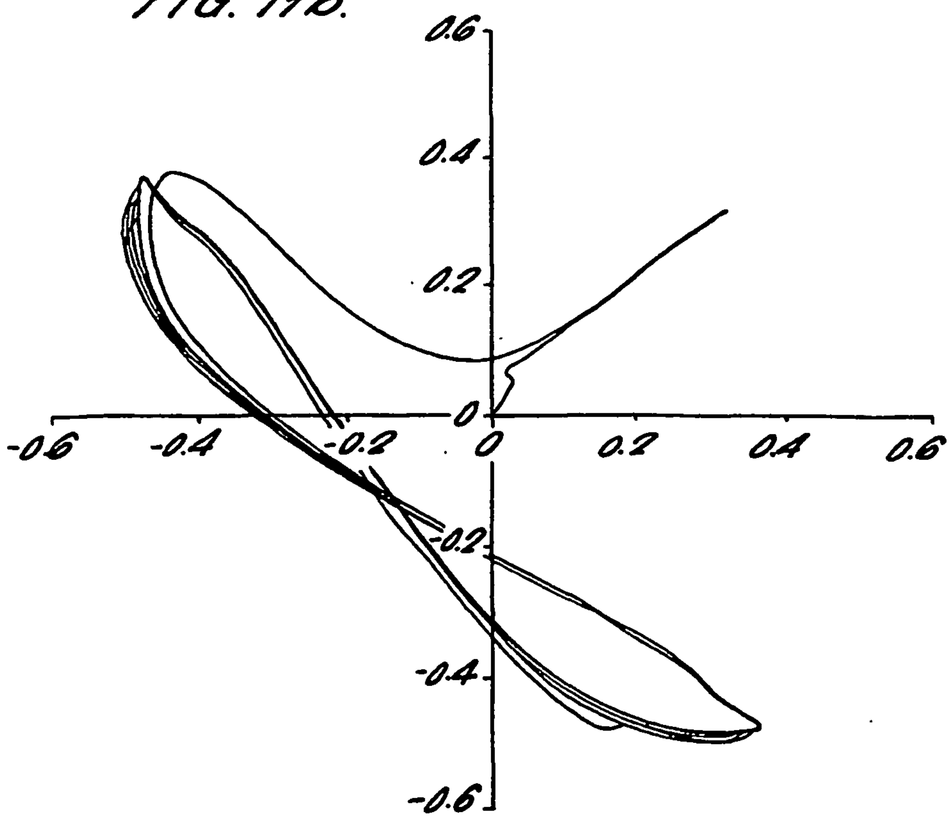


FIG. 12.

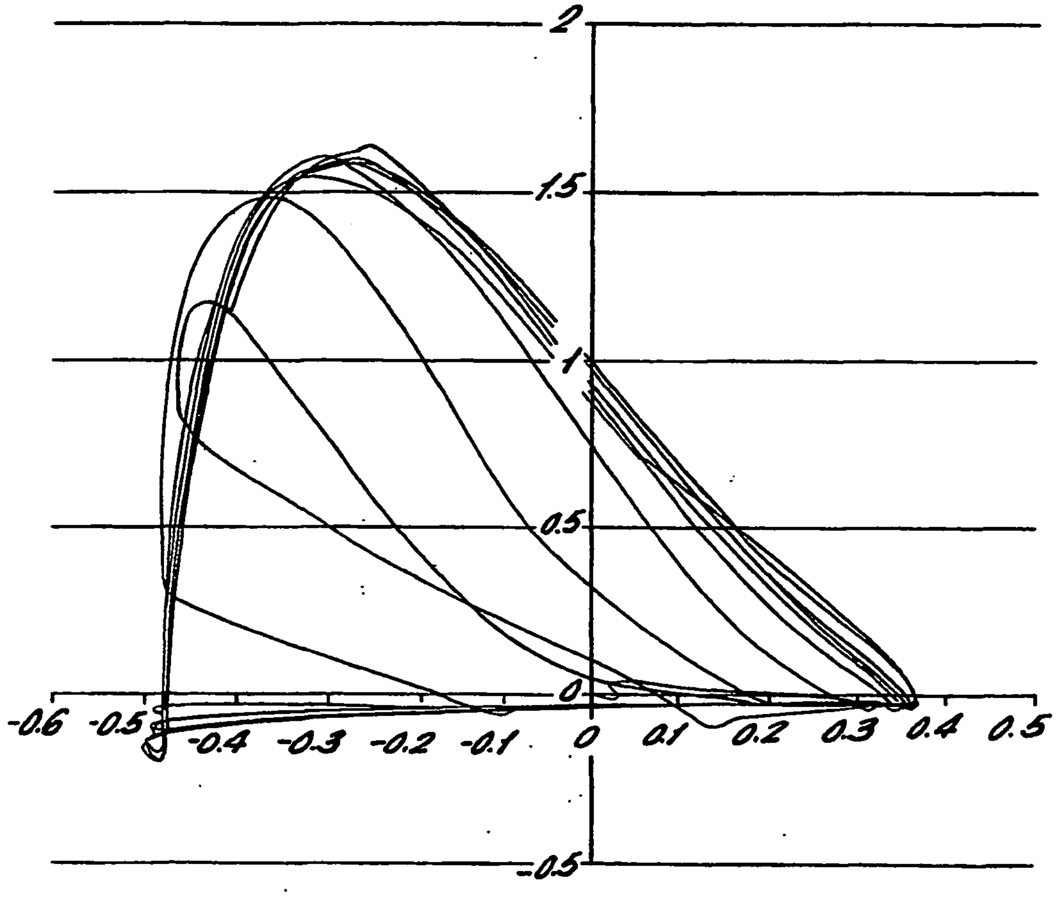
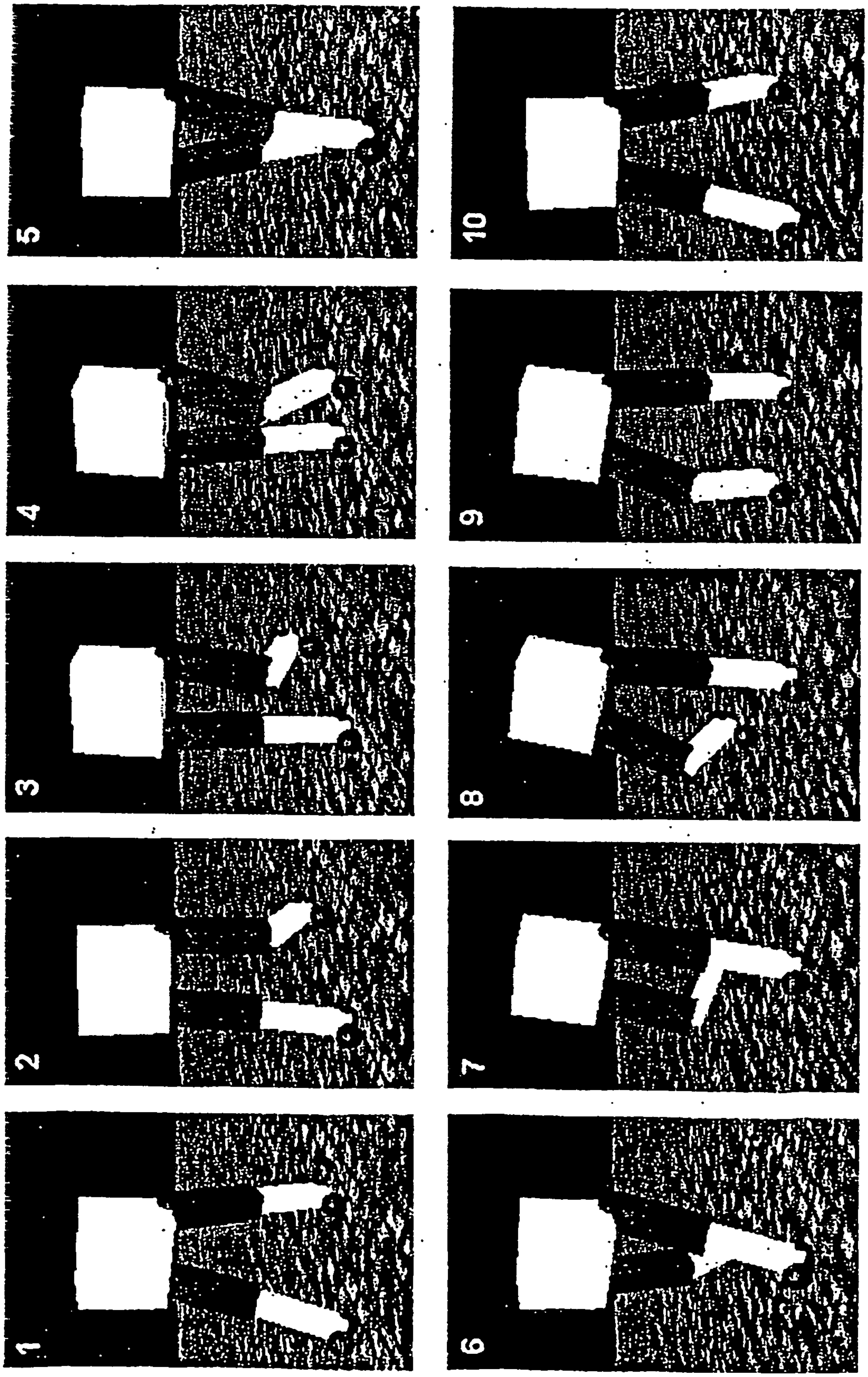


FIG. 13.



REFERENCES CITED IN THE DESCRIPTION

Patent documents cited in the description

- US 5124918 A **[0005]**

Non-patent literature cited in the description

- From Lampreys to Salamanders: Evolving Neural Controllers for Swimming and Walking. **IJSPEERT et al.** Animals to Animats, Proceedings of the 5th International Conference of the Society for Adaptive Behaviour (SAB98. MIT Press, 1998, 390-399 **[0005]**
- **GOLUBITSKY et al.** Symmetry in locomotor central pattern generators and animal gaits. *Nature,* 1999, vol. 401, 693-695 **[0005]**
- **J. E. PRATT et al.** Exploiting Natural Dynamics in the Control of a 3D Biped Walking Simulation. *International Conference on Climbing and Walking Robots,* September 1999 **[0007]**
- Simulation of Autonomous Biped Walking Robot Including Environmental Force Interaction. *IEEE Robotics & Automation Magazine,* 01 June 1998 **[0008]**
- **Y BELLAN et al.** Artificial Neural Networks for Motion Emulation in Virtual Environments. *Proceedings of the International Workshop on Modelling and Motion Capture Techniques,* 1998 **[0009]**
- Autonomous Trajectory Generation of a Biped Locomotive Robot. *Proceedings of the International Joint Conference on Neural Networks,* 18 November 1991 **[0010]**
- Co-evolving parasites improves simulated evolution as an optimisation procedure. **W.D. HILLIS et al.** Artificial Life II. Addison-Wesley, 313-324 **[0075]**